

KLASIFIKASI JATUH MENGGUNAKAN ALGORITMA DECISION TREE

VerinRefinda Dwinanda Danandjojo

Jurusan Teknik Informatika, Fakultas Teknologi Industri
Universitas Islam Indonesia
verin1993@gmail.com

I. ABSTRAKSI

Kecelakaan dan cedera yang diakibatkan oleh terjatuh biasanya lebih rentan terjadi pada lansia. Banyak pencegahan jatuh dengan menggunakan pendekatan-pendekatan yang telah dikembangkan berkisar dari pengobatan dan penyesuaian lingkungan, menggunakan pengekang fisik sampai bantalan pelindung pinggul dan sistem komersial semacam perlindungan diri untuk mengurangi resiko dari terjatuh. Kecelakaan jatuh lebih sering dialami oleh lansia yang sudah berkurang optimalisasi kinerja tubuh, yang mengakibatkan kondisi dimana mereka mudah untuk terjatuh. Kurangnya pengawasan dan keterbatasan fisik para lansia yang sulit untuk meminta bantuan menjadi dasar gagasan dibuatnya alat pendeteksi jatuh dengan dasar accelerometer. Alat pendeteksi jatuh ini akan dibawa oleh objek di dalam kantung mereka. Kinerja nya alat ini akan memberikan laporan atau peringatan kepada sanak saudara atau orang yang dihibau oleh objek bilamana bergerak alat ini saat dibawa oleh objek mengalami pergerakan jatuh, yaitu pergerakan dari posisi semula ke posisi terjatuh kebawah. Namun keakurasian pendeteksi jatuh pada saat kejadian menjadi masalah yang cukup besar. Setelah diselidiki kinerja berbagai pengklasifikasi pada pendeteksi jatuh, *Threshold* yang didapat sensitivitas tertinggi hanya mencapai 90,15%. Perbedaan hasil deteksi dengan data yang telah di *record* dapat menjadikan *report* yang diberikan oleh pendeteksi jatuh tidak sesuai dengan apa yang terjadi dilapangan. Untuk mengatasi kurangnya keakurasian pada saat mendeteksi maka dilakukan penelitian ini untuk mencari cara menambah keakurasian pendeteksi jatuh pada saat mendeteksi. Diawali dengan kita ambil data sampel untuk diuji coba disini kita ambil setelah melalui *preprocessing*, *windowing*, *ekstraksi data*, dan *seleksi fitur* kami dapat 140 *dataset* yang mewakili prediksi *activity falling*, *lying*, *lying down*, *sitting*, *standing up from lying*, *standing up from sitting*, dan *walking*. Masing masing kategori posisi tersebut diwakili 20 *dataset*. Data sampel yang telah diambil akan dihitung dengan algoritma *decision tree* menggunakan RapidMiner. Sebelum diolah *dataset* dibagi menjadi dua kategori terlebih dahulu, yaitu data MIN (nilai data di bawah nilai pengkategorian) dan data MAX (nilai data di atas nilai pengkategorian). *decision tree* yang digunakan information gain dan tanpa menggunakan pruning. Hasil perhitungan *decision tree* menggunakan RapidMiner selanjutnya diuji keakurasian dengan menggunakan metode *confusion matrix*. Data hasil yang diambil ada tiga yaitu bagan *decision tree* tabel klasifikasi data dan tabel validasi. Dari hasil perhitungan dan validasi didapat *sensitifity* 82%, *spesifity* 97%, dan *accuracy* 95%. Untuk mengetahui ketepatan

pengujian, dilakukan perhitungan manual untuk *splitter* pertama pada *decision tree* dan *confusion matrix* pada *activity falling*. Perhitungan atau pemetaan *decision tree* menggunakan RapidMiner dengan perhitungan manual menemukan kecocokan pada *splitter* pertama dan variable dengan Gain terbesar yaitu *std_x*. sedangkan pada perhitungan *confusion matrix* masih memiliki perbedaan. Perhitungan menggunakan RapidMiner *accuracy falling* mencapai 81.43%, sedangkan *accuracy falling* pada perhitungan manual mencapai 99%.

Kata kunci: jatuh, klasifikasi, *decision tree*.

II. PENDAHULUAN

Setiap orang mengalami kejadian terjatuh yang tidak diinginkan dari kecil hingga dewasa. Terjatuhpun dapat dimanfaatkan oleh beberapa orang meski demikian, sulit untuk menggambarkan secara tepat fenomena tersebut, dan bahkan lebih sulit lagi membayangkan sarana untuk pendeteksinya. Jatuhnya seseorang dapat digambarkan sebagai pergerakan cepat berubah dari satu posisi keposisi yang lain, dari posisi tegak atau posisi duduk ke posisi berbaring atau hampir terbaring (posisi ini bukan gerakan terkontrol). Semisal pada tahun 1987 kelompok kerja internasional Kellogg dalam pencegahannya Jatuh pada orang tua didefinisikan sebagai "tidak sengaja menuju ke arah tanah, atau beberapa tingkat yang lebih rendah bukan sebagai konsekuensinya menahan pukulan keras, kehilangan kesadaran, mendadak kelumpuhan seperti pada stroke atau epilepsi". Menurut definisi tersebut detektor jatuh harus dapat mendeteksi terjatuh ke tanah, tak terduga dengan cara yang tidak terkendali, apapun penyebab kecelakaan jatuh tersebut.

Kesulitan bangkit kembali setelah terjatuh dialami oleh orang dewasa yang lebih tua. Hanya 49% dari orang tua terjatuh yang dapat bangkit kembali tanpa bantuan, dan sebagian besar orang tua yang terjatuh tidak memiliki kemampuan untuk bangkit kembali tanpa mendapatkn bantuan dari orang lain tanpa terkait cedera serius. Ketidakmampuan untuk bangun setelah jatuh bias juga diakibatkan cedera yang berhubungan dengan jatuh. Orang tua yang terjatuh dan tidak memiliki kemampuan untuk bangkit akan lebih buruk bila tidak ada orang yang membantu untuk bangun dalam jangka waktu yang lama, rasa sakit yang diterima akan diperburuk dengan kemungkinan adanya dehidrasi, luka tekanan, dan atau cedera otot. (P.Rumeau, Bourke, OLaighin, Lundy, & Noury, 2008).

Sistem deteksi jatuh sangat diperlukan paramedis agar dapat memberikan penanganan medis yang lebih

cepat. Sistem ini menjadi penting karena jatuh merupakan salah satu permasalahan utama pada masyarakat dengan tingkat populasi manula yang tinggi. Kecelakaan jatuh menjadi penyebab kematian terbesar ke-6 bagi orang berusia di atas 65 tahun, ke-2 bagi orang pada usia 65 hingga 75 tahun, dan penyebab kematian terbesar pada orang berusia di atas 75 tahun (Khawandi, Daya, & Chauvet, 2011).

Sistem deteksi ini akan mempermudah kita untuk mengurangi dampak buruk dari terjatuhnya orang dewasa yang lebih tua. Saat kita telah mengetahui kapan dan seperti apa kondisi yang memiliki kemungkinan lebih besar untuk terjatuh bagi orang dewasa yang lebih tua disekitar kita, akan lebih mudah untuk mengawasi atau mengantisipasi.

Berdasarkan latar belakang tersebut maka dapat diambil rumusan masalah bagaimana membangun sistem deteksi jatuh menggunakan Decision Tree. Tujuan dari penelitian ini adalah dapat membantu mengukur keakuratan deteksi kejadian jatuh dengan menggunakan metode *Decision Tree*.

III. LANDASAN TEORI

Jatuh merupakan kejadian yang melibatkan seseorang mendadak terbaring, terduduk dilantai, berada ditempat yang lebih rendah, dengan tanpa sadar dan atau menimbulkan luka. Pada tahun 1987 kelompok kerja internasional Kellog melakukan pencegahan jatuh pada orang tua didefinisikan jatuh sebagai “ tidak sengaja datang ke tanah, atau beberapa tingkat yang lebih rendah bukan sebagai konskuensiannya menahan pukulan keras, kehilangan kesadaran, dan seketika mengalami kelumpuhan seperti pada stroke atau serangan epilepsi”. Definisi ini digunakan pada banyak penelitian, dan kemudian diperluas termasuk jatuh akibat pusing dan lainnya. Jatuh sering terjadi atau dialami oleh usia lanjut.

Kecelakaan dan cedera yang diakibatkan oleh terjatuh biasanya lebih rentang terjadi pada lansia. Banyak pencegahan jatuh dengan menggunakan pendekatan-pendekatan yang telah dikembangkan berkisar dari pengobatan dan penyesuaian lingkungan, menggunakan pengekang fisik sampai bantalan pelindung pinggul dan sistem komersil semacam perlindungan diri untuk mengurangi resiko dari terjatuh.

Ukuran ini berdasarkan pada karya Claude Shannon pada teori informasi, yang mempelajari nilai atau “isi informasi” dari sebuah pesan. Biarkan node mewakili atau menahan tupel partisi D. atribut dengan gain informasi tertinggi dipilih sebagai atribut pemisah node. Atribut ini meminimalisir informasi yang dibutuhkan untuk mengklasifikasikan tupel dalam partisi yang dihasilkan dan mencerminkan keacakan atau ketidakamanan terkecil dalam partisi ini. Pendekatan ini meminimalkan jumlah tes yang diharapkan yang diperlukan untuk mengklasifikasikan tupel yang diberikan dan menjamin bahwa tree yang sederhana (tapi tidak harus yang paling sederhana) dapat ditemukan.

Informasi yang diharapkan untuk mengklasifikasi tupel di D didapat dari

$$Info(D) = - \sum_{i=1}^m p_i \log_2(p_i) \quad (2.1)$$

Dimana p_i adalah probabilitas tak berhingga bahwa tuple acak di D termasuk kelas C_i dan diperkirakan oleh $|C_i, D|/|D|$. Fungsi log ke basis 2 digunakan, karena informasinya dikodekan dalam bit. Info (D) hanyalah jumlah rata-rata informasi yang dibutuhkan untuk mengidentifikasi label kelas tupel di D. informasi yang kita miliki didasarkan pada proporsi tupel dari setiap kelas info (D) juga dikenal sebagai entropi dari D.

Beberapa banyak lagi informasi yang masih kita butuhkan (setelah partisi) untuk sampai pada klasifikasi yang tepat jumlah ini diukur dengan

$$INFO_A(D) = \sum X INFO(D_i) \quad (2.2)$$

Syarat $\frac{|D_j|}{|D|}$ bertindak sebagai bobot partisi j th. $info_A(D)$ adalah informasi yang diharapkan diperlukan untuk mengklasifikasikan tuple dari D berdasarkan partisi oleh A. semakin kecil informasi yang diharapkan (masih) dibutuhkan, semakin besar kemurnian partisi. Keuntungan informasi didefinisikan sebagai perbedaan antara informasi asli persyaratan (yaitu, hanya berdasarkan proporsi kelas) dan persyaratan baru (yaitu, diperoleh setelah partisi pada A). itu adalah

$$GAIN(A) = INFO(D) - INFO_A(D) \quad (2.3)$$

Dengan kata lain, Gain(A) memberitahukan kita ada berapa banyak yang akan diperoleh dengan bercabang pada A. ini penurunan yang diharapkan dalam kebutuhan informasi yang disebabkan oleh mengetahui nilai A. atribut A dengan gain(A), dipilih sebagai pemecahan atribut pada node N ini setara dengan mengatakan bahwa kita ingin mempartisi atribut tersebut A yang akan melakukan “klasifikasi terbaik”, sehingga jumlah informasi masih diperlukan untuk menyelesaikan pengklasifikasian tuple minimal (yaitu, $info_A(D)$).

Validasi dimungkinkan untuk mendisain pekerjaan eksperimen sedemikian rupa agar sesuai dengan karakteristik dapat memberikan pengujian secara keseluruhan data mengenai prosedur analitis, seperti specificity (spesifikasi), linearity (linearitas), range (jarak), accuracy (akurasi) dan precision (presisi). Tujuan dari prosedur analitik harus dipahami dengan tujuan yang jelas karena akan mengatur karakteristik validasi yang perlu dievaluasi.

Accuracy

Keakuratan prosedur analitik menyatakan kedekatan kesepakatan antara nilai yang diterima baik sebagai nilai nyata konvensional atau nilai referensi yang diterima dan nilai yang ditemukan. Accuracy terkadang disebut trueness. Untuk menghitung akurasi dapat menggunakan rumus berikut :

$$Accuracy = \left(Sensitivity \times \frac{pos}{pos+neg} \right)_+ \quad (2.6)$$

$$: \left(Sensitivity \times \frac{neg}{pos+neg} \right)$$

Precision

Ketepatan prosedur analitik menyatakan kedekatan perjanjian (tingkat scatter) antara serangkaian pengukuran yang diperoleh dari beberapa contoh yang sama dalam kondisi yang

ditentukan. Presisi dapat dipertimbangkan pada tiga tingkat : pengulangan, presisi antara dan reproduktifitas. Preasisi seharusnya diselidiki menggunakan contoh yang homogeny dan ontetik dengan menggunakan ruus berikut :

$$\text{Precision} = \frac{t_pos}{t_pos + f_pos} \quad (2.7)$$

Namun jika itu tidak mungkin diperoleh sampel yang homogen dapat diselidiki menggunakan sampel yang dibuat secara artifisial atau larutan sampel. Ketepatan prosedur analitik biasanya dinyatakan sebagai varians, standar deviasi atau koefisien variasi dari serangkaian pengukuran

Specificity

Spesifisitas adalah kemampuan untuk menilai dengan tegas analit di hadapan komponen yang mungkin diharapkan hadir dengan menggunakan rumus berikut :

$$\text{Specifity} = \frac{t_neg}{neg} \quad (2.8)$$

Biasanya ini mungkin termasuk kotoran, degradasi, matriks dan lainnya. Kekurangan kekhususan prosedur analitis individu dapat dikompensasi oleh prosedur analitis individu dapat dikompensasi oleh prosedur analitis pendukung lainnya.

IV. METODOLOGI

A. Ekstraksi Fitur

Fitur ini merupakan fungsi yang dapat mengukur berbagai aspek dataset dan nilainya dapat menjadi ciri kumpulan data. Sementara ekstraksi fitur dapat didefinisikan sebagai sebuah proses dalam menganalisa dan kemudian mengubah dataset menjadi fitur database yang berarti untuk diproses menjadi langkah selanjutnya. Bagian ini mengenalkan jenis fitur yang diambil dari kumpulan data log accelerometer. Dapat tercerminkan perilaku atau pergerakan pengguna sesuai dengan keadaan dan kegiatan sehari hari. Tujuannya adalah untuk mengekstrak parameter yang mewakili informasi diskriminatif (MDPI, 2014), misalkan setiap orang sering melakukan kegiatan duduk, berjalan, dan tiarap dengan komposisi dan keadaan yang berbeda-beda. Setiap pengguna mungkin memiliki tempat kunjungan yang berbeda, bentuk tubuh yang berbeda-beda, lingkungan yang berbeda, aktivitas yang berbeda, dan lain-lain. Berdasarkan Tabel 1 Rumus fitur, fitur diambil dari sinyal akselerasi, seperti mean, standar deviasi, jumlah besaran vector, dan sudut kemiringan sederhana, namun kurang 3nformative untuk fitur yang umum digunakan dalam eksperimen pendeteksi jatuh yang ada. Persamaan generic seperti mean, standar deviasi dan variasi dapat diterapkan pada sensor gerak dan oleh karena itu input dari persamaan dilambangkan dengan tanda x.

Fitur dan rumus-rumus untuk percobaan jatuh yaitu :

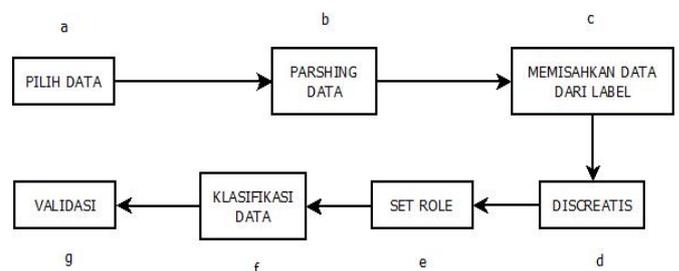
Tabel 1 Rumus fitur

NO	FITUR	RUMUS
F1	Mean	$\mu = \frac{1}{N} \sum_{i=1}^N x_i$
F2	Standar Deviasi	$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$
F3	Variance	$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2$

F4	Standard deviation magnitude	$ \sigma = \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2}$
F5	Sum vector magnitude	$ a = \sqrt{a_x^2 + a_y^2 + a_z^2}$
F6	Sum vector on horizontal plane	$ a _h = \sqrt{a_x^2 + a_y^2}$
F7	Standard deviation of sum vector magnitude	$\sigma_{ a } = \sqrt{\frac{1}{N} \sum_{i=1}^N (a _i - \mu_{ a })^2}$
F8	Difference between and maximum and minimum values of sum vector magnitude	$\Delta a _{max-min} = \max(a) - \min(a)$
F9	Root mean square of sum vector magnitude	$ a _{max-min} = \sqrt{\frac{1}{N} \sum_{i=1}^N a _i^2}$
F10	Signal magnitude area	$SMA = \frac{1}{t} \left(\int_0^t a_x(t) dt + \int_0^t a_y(t) dt + \int_0^t a_z(t) dt \right)$
F11	Activity signal magnitude area	$ASMA = \frac{1}{t_2 - t_1} \left(\int_{t_1}^{t_2} \sqrt{a_x^2(t) + a_y^2(t) + a_z^2(t)} dt \right)$

B. Proses Klasifikasi

Sistem ini menggunakan algoritma decision tree, Algoritma decision tree adalah sebuah struktur data yang terdiri dari simpul(node) dan rusuk(edge). Pohon keputusan ini dapat memberikan keuntungan berwujud visualisasi dari pemecah masalah yang diolah menggunakan teknik data mining yang membuat aturan dari prediksinya dapat diamati, maka dari itu konsep ini termasuk fleksibel dan antraktif. Pada pohon keputusan ini kamu menggunakan algoritma C4.5, karena dasar algoritma C4.5 adalah pohon keputusan. Algoritma data mining C4.5 merupakan salah satu algoritma yang digunakan untuk melakukan klasifikasi atau segmentasi atau pengelompokan yang bersifat prediktif. Cabang-cabang pohon keputusan merupakan pertanyaan klasifikasi dan daun-daunnya merupakan kelas-kelas atau segmen-segmennya. Pada penelitian ini algoritma klasifikasi akan dilakukan menggunakan RapidMiner. Data yang telah siap akan dihitung menggunakan algoritma decision tree dan akan memberikan data hasil berupa bagan decision tree dan bagan Confusion Matrix. Berikut tahapan-tahapan klasifikasi menggunakan RapidMiner :



Gambar IV.1 Proses klasifikasi

1) PILIH DATA :

Setelah kita masuk pada halaman kerja, karena masih kosong maka akan kita isi dengan data dan operator-operator yang kita butuhkan. Pertama-tama kita masukkan terlebih dahulu data kita yang telah siap. Untuk memasukkan data, kita masukan

melalui bar Add Data pada kolom Repository. Mula-mula kita memilih akan mengambil data dari database RapidMiner yang sudah ada atau dari komputer kita. Dalam kasus ini kita mengambil data yang disimpan di komputer.

2) **PARSHING DATA** :

Setelah kita memilih data RapidMiner akan memberikan review awal data kita, pada review ini kita pun dapat menentukan spesifikasi data kita sesuai dengan kebutuhan. Mulai dari menentukan header dari tiap kolom atau baris, menentukan kolom separator hingga encoding file yang digunakan. Setelah spesifikasi data telah ditentukan selanjutnya kita menentukan format dari tiap kolom dan data yang akan digunakan.

3) **MEMISAHKAN LABEL DATA** :

Sebelum diklasifikasikan data terlebih dahulu dipisahkan dari label masing-masing data agar kolom label data tidak ikut serta terhitung dalam pengklasifikasian dan memberikan hasil yang maksimal.

4) **DIZCRETIZE** :

Klasifikasi langkah pertama melakukan pembagian dari keseluruhan menjadi dua atau empat kelompok sesuai dengan data agar lebih mudah untuk menentukan tiap data akan masuk ke dalam kategori yang mana.

5) **KLASIFIKASI DATA** :

data diklasifikasikan menggunakan algoritma decision tree sesuai dengan spesifikasi yang telah ditentukan.

6) **VALIDASI** :

Hasil klasifikasi menggunakan decision tree akan diproses sehingga menghasilkan bagan decision tree, menjadi data pengkategorian sehingga dapat dihitung presentasi keakurasiannya menggunakan confusion matrix.

C. **Algoritma klasifikasi**

Sistem ini menggunakan algoritma *decision tree*, Algoritma decision tree adalah sebuah struktur data yang terdiri dari simpul(node) dan rusuk(edge). Pohon keputusan ini dapat memberikan keuntungan berwujud visualisasi dari pemecah masalah yang diolah menggunakan teknik data mining yang membuat aturan dari prediksinya dapat diamati, maka dari itu konsep ini termasuk fleksibel dan antraktif. Pada pohon keputusan ini kamu menggunakan algoritma C4.5, karena dasar algoritma C4.5 adalah pohon keputusan. Algoritma data mining C4.5 merupakan salah satu algoritma yang digunakan untuk melakukan klasifikasi atau segmentasi atau pengelompokan yang bersifat prediktif. Cabang-cabang pohon keputusan merupakan pertanyaan klasifikasi dan daun-daunnya merupakan kelas-kelas atau segmen-segmennya.

Rumus menghitung entropy pada algoritma C4.5 :

$$ENTROPY(S) = - \sum_{i=1}^n P_i \times \log_2 P_i \tag{3.1}$$

Keterangan :

- S : himpunan (dataset) kasus
- K : banyaknya partisi S
- Pi : probabilitas yang didapat dari Sum(Ya) atau Sum(Tidak) dibagi total kasus

Setelah mendapatkan entropi dari keseluruhan kasus, lakukan analisis pada setiap atribut dan nilai-nilainya dan hitung entropinya. Langkah berikutnya yaitu dengan menghitung Informasi Gain, rumus menghitung Informasi Gain sebagai berikut :

$$INFORMASI\ GAIN(S) = ENTROPY(S) - \sum_{i=1}^n \frac{n_i}{n} \times ENTROPY(S_i) \tag{3.2}$$

Keterangan :

- S : Himpunan kasus
- n : Jumlah partisi dalam S
- Si : Partisi dari S

V. **HASIL DAN PEMBAHASAN**

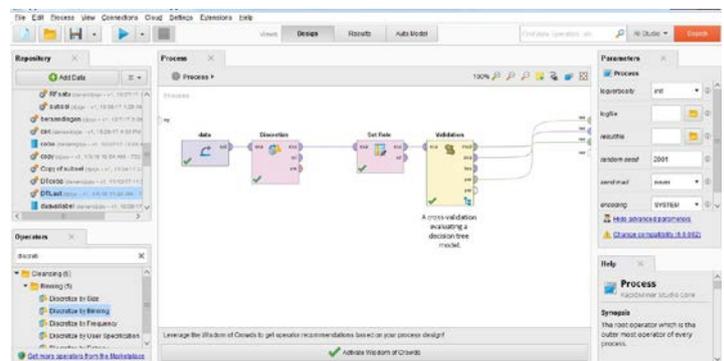
Jatuh adalah kejadian yang melibatkan seseorang mendadak terbarik, terduduk dilantai, berada ditempat yang lebih rendah, dengan tanpa sadar dan atau menimbulkan luka. Dari penjelasan diatas, mengartikan bahwa saat seseorang terjatuh bisa berada pada posisi yang berbeda-beda. Maka penelitian ini mengambil data contoh dari terjatuh yang memiliki posisi akhir yang berbeda-beda.

Terdapat 140 dataset yang telah di olah sehingga dapat digunakan dan dibagi menjadi tujuh kategori data sesuai dengan posisi akhir jatuh pada saat mengambil setiap dataset.

Data-data tersebut kami kelompokkan sesuai dengan kategori dari posisi akhir setiap percobaan dengan nilai variabelnya masing-masing seperti pada Tabel 4.1 Dataset. Tampak pada setiap dataset memiliki prediksi masing-masing. Prediksi tersebut diambil dari nilai variable-variabel yang menggambarkan prediksi posisi activity.

A. **Perhitungan Decision Tree**

RapidMiner merupakan perangkat lunak bersifat terbuka (open source) yang menjadi sebuah solusi untuk melakukan analisis terhadap data mining, text mining dan analisis prediksi. Data yang sudah kami bagi berdasarkan posisi terakhir objek pada saat pengujian. Hasil pengujian tersebut akan kami hitung ulang menggunakan metode decision tree, untuk membuktikan ketepatan prediksi atau hasil pengujian berdasarkan nilai titik koordinat setiap pengujian.



gambar V.1 Proses

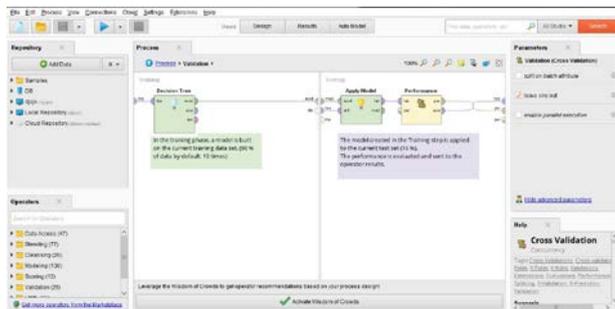
Berdasarkan gambar V.1 Proses ada beberapa operator yang digunakan. Berikut keterangan dan fungsi dari operator-operator tersebut :

Retrieve (data) : Terdapat 140 data sampel yang telah disiapkan untuk diklasifikasikan dengan metode decision tree yang hasilnya akan diolah lagi untuk menentukan keakurasian pengkategorian menggunakan confusion matrix.

Discretize (by binning) : data awal dibagi atau di diskrit sesuai dengan atribut numeric dari masing-masing data. Pembagian yang dilakukan berdasarkan jumlah parameter bins sehingga dapat menentukan jumlah bins yang diperlukan. Data ini akan di diskrit dengan mengacu pada rentang nilai masing – masing kelompok data dengan batasan min(nilai rentang terendah) dan max(nilai rentang tertinggi) sebagai batasan dari masing-masing bins.

Set Role : setiap data yang ada terbagi menjadi beberapa baris data prediksi sesuai posisi pada saat diambilnya data tersebut, masing-masing data prediksi juga memiliki kolom atribut koordinat sesuai dengan kasus. Set Role berfungsi untuk membedakan baris penamaan atribut koordinat dan prediksi posisi yang akan di masukan kedalam kategori ‘label’. Agar pada saat pengkategorian data ‘label’ tidak ikut serta terhitung dan merubah hasil.

Validation (Cross Validation) : setelah data melewati beberapa operator untuk mempersiapkan data, operator validasi memulai proses pengkategorian. Terbagi menjadi dua subproses Training dan subproses Testing. Pada subproses Training diletakan operator decision tree, data akan diuji atau diproses dengan menggunakan algoritma decision tree. Selesai diproses data hasil lalu masuk kedalam subproses Testing untuk diukur hasil kinerjanya.



gambar V.2 Validasi

Di dalam Validasi seperti pada gambar V.2 Validasi terdapat beberapa operator pendukung sebagai berikut :

Modeling (Decision tree) : operator model ini menerapkan algoritma decision tree dengan atribut model yang telah disesuaikan, dipilih decision tree dengan metode information gain.

Apply Model : Operator ini akan menguji data hasil dari operator decision tree setelah itu, model ini mendapatkan prediksi pada data yang tidak terlihat atau untuk mengubah data dengan menerapkan model preprocessing.

Performance : pada hasil akhir proses ini performance akan menampilkan hasil. Data hasil pembagian masing-masing data sesuai prediksi atau tidak hingga presentasi keakurasian pengkategorian decision tree,

dengan memberikan dua hasil pokok yaitu bagan decision tree dan tabel performance.

Hasil performance vector pada gambar dan gambar menunjukkan seberapa besar ketepatan ataupun kesamaan hasil data percobaan jatuh dengan perhitungan decision tree, yang dalam kasus ini kami menggunakan decision tree information gain.

gambar V.3 Performance

Tampak pada gambar V.3 Performance hasil diatas bahwa ketepatan akurasi decision tree dengan hasil uji coba jatuh mencapai 81.43% . performance vector juga memberitahukan berapa dan pada kategori mana saja data yang sesuai atau tidak sesuai antara hasil ujicoba dengan perhitungan decision tree.

Dalam hal ini hasil perhitungan yang di dapat dikategorikan sesuai dengan ketepatan prediksinya menjadi true positive, true negative, false positive, dan false negative . true positive adalah jumlah hasil positif yang diklasifikasikan sebagai positif, false positive adalah jumlah hasil negatif yang diklasifikasikan sebagai positif, false negative adalah jumlah hasil positif yang diklasifikasikan negatif, true negative adalah jumlah hasil negatif yang diklasifikasikan sebagai negatif.

Tabel 2 Performance vector

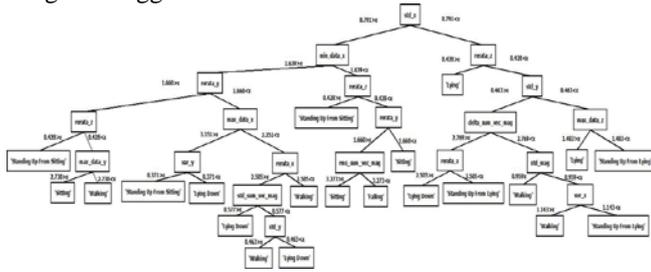
Kategori	True Positive	False Positive	False Negative	True Negative
Falling	20	1	0	119
Lying	16	5	3	116
Lying Down	18	5	2	115
Sitting	16	2	4	118
Standing up from Lying	12	6	8	114
Standing up from Sitting	16	1	4	119
walking	16	6	4	114

Hasil perhitungan Tabel 2 Performance vector yang dihitung dengan teori decision tree informasi gain bila dikelompokkan true positive, true negative, false positive, dan false negative dari setiap kategori maka akan menjadi seperti pada tabel. Falling menjadi kelompok kategori yang paling akurat.

B. Perhitungan Manual

Pada bagian ini kita akan mencoba mencoba membuktikan perhitungan dari hasil uji coba menggunakan RapidMiner,

akan ada dua pembuktian yang pertama pembuktian penentuan decision tree dan yang kedua menghitung hasil validasi dengan menggunakan metode confusion matrix.



gambar V.4 Bagan decision tree

hasil ini menjelaskan pemetaan pada Gambar 4.4.2.1 Hasil decision tree yang dilakukan oleh RapidMiner dengan data yang telah di masukkan. Dari gambar hasil decision tree di awali dengan root pertamanya *std_x* . perhitungan ini menggunakan metode decision tree Informasi Gain.

Untuk membuktikan perhitungan decision tree oleh RapidMiner kami akan mencoba menghitung ulang secara manual, dengan rumus decision tree informasi gain.

Agar dapat dihitung data-data yang tadi sudah terbagi menjadi tujuh kategori falling, lying, lying down, sitting, standing up from lying, standing up from sitting, dan walking, dibagi lagi menjadi dua jenis yaitu data MIN (nilai data di bawah nilai pengkategorian) dan data MAX (nilai data di atas nilai pengkategorian) .

Data-data lalu dihitung menggunakan rumus decision tree rapidminer sebagai berikut (Han, Kammer, & Pei, 2012):

$$\text{Entropy}(S) = \sum_{i=1}^n -p_i \times \log_2 p_i \quad (4.1)$$

Keterangan :

- S : Himpunan kasus
- n : Jumlah partisi dalam S
- Pi : Partisi dari Si terhadap S

$$\text{Informasi Gain}(S) = \text{Entropy}(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} \times \text{info}(S_i) \quad (4.2)$$

Keterangan :

- S : Himpunan kasus
- n : Jumlah partisi dalam S
- Si : Partisi dari S

Untuk menentukan root pertama kita cari data yang memiliki nilai information gain tertinggi, perhitungan dengan menggunakan rumus-rumus diatas mendapatkan hasil nilai information gain tertinggi pada data *std_x* dengan nilai 0.74144313. hasil perhitungan manual dan perhitungan rapid miner dalam menentukan root pertama sesuai atau sama yaitu pada variable *std_x*.

Bila menggunakan tabel matriks, dataset yang diuji dianggap sebagai positif dan yang lainnya negative. Dari Tabel 4.1.1 Performance vector yang telah membagikan data hasil menjadi True Positive, False Positive, True Negative, dan False Negative dapat diketahui sensitifity(recall), specifity, precision, dan accuracy. Menggunakan perhitungan sebagai berikut (Han, Kammer, & Pei, 2012):

$$\text{Sensitifity(recall)} : \frac{t_pos}{pos} \quad (4.3)$$

$$\text{Specifity} : \frac{t_neg}{neg} \quad (4.4)$$

$$\text{Precision} : \frac{t_pos}{t_pos + f_pos} \quad (4.5)$$

$$\text{Accuracy} : \left(\text{Sensitifity} \times \frac{pos}{pos+neg} \right) + \left(\text{Sensitifity} \times \frac{neg}{pos+neg} \right) \quad (4.6)$$

Keterangan :

- t_pos : jumlah true positive
- t_neg : Jumlah true negative
- pos : Jumlah record positive
- Neg : Jumlah record negative
- f_neg : Jumlah false positive

Dari rumus diatas dilakukan perhitungan confusion matrix falling berdasarkan data dari Tabel 2 Performance vector :

Tabel 3 Falling

Falling	True Positive	False Positive	False Negative	True Negative
	20	1	0	119

$$\text{Sensitifity} : \frac{20}{20} = 1 \times \frac{100}{100} = 100\%$$

$$\text{Specifity} : \frac{119}{120} = 0.99 \times \frac{100}{100} = 99\%$$

$$\text{Precision} : \frac{20}{20+1} = \frac{20}{21} = 0.95 \times \frac{100}{100} = 95\%$$

$$\begin{aligned} \text{Accuracy} &: \left(1 \times \frac{20}{20+120} \right) + \left(0.99 \times \frac{120}{20+120} \right) = \\ &= (1 \times 0.143) + (0.99 \times 0.857) = 0.99 \\ &= 0.99 \times \frac{100}{100} = 99\% \end{aligned}$$

Hasil perhitungan diatas lalu kami ubah ke dalam bentuk tabel seperti Tabel 4 confusion matrix sebagai berikut :

Tabel 4 confusion matrix

Confusion Matrix	Falling
Sensitifity	100 %
Specifity	99%
Precision	95%
Accuracy	99%

Perhitungan accuracy deteksi jatuh dengan algoritma decision tree menggunakan RapidMiner memiliki nilai 81.43%. berbeda dengan hasil perhitungan confusion matrix secara manual yang dapat mencapai hasil accuracy 99%. Hasil ini menggambarkan meningkatnya accuracy falling dengan menggunakan algoritma decision tree.

VI. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil penelitian pada klasifikasi jatuh menggunakan decision tree dapat diambil beberapa kesimpulan sebagai berikut :

- a. Klasifikasi jatuh dengan decision tree dapat mengklasifikasikan data percobaan falling, lying, lying down, sitting, standing up from lying, standing up from sitting, dan walking.
- b. Terdapat perbedaan perhitungan akurasi deteksi jatuh, Akurasi klasifikasi jatuh menggunakan decision tree (Informasi Gain) mencapai 81.43% sedangkan hasil perhitungan manual dengan menggunakan metode Confusion Matrix mencapai 99%.
- c. Dari hasil performance vector hasil percobaan pada kategori falling memiliki keakurasian tertinggi dengan 100%, dan standing up from lying memiliki keakurasian terendah dengan 66.67% .
- d. Berdasarkan keakurasian laporan pada posisi falling lebih akurat dari pada pada kategori lainnya.

B. Saran

Saran yang kami ajukan dari hasil penelitian yaitu sebagai berikut :

- a. Terlalu banyak nya pengkategorian, jika hanya menggunakan dua kategori data masih dapat diuji coba menggunakan metode confusion matrix untuk mengukur keakurasian.

- b. Kurang banyak nya data sampel yang digunakan.

VII. REFERENCES

- [1] carthheel technologies. (2018, May 2). Retrieved from Binning: <http://www.saedsayad.com/binning.htm>
- [2] Han, J., Kammber, M., & Pei, J. (2012). *DATA MINING Concepts and Techniques*. Wyman Street, Waltham: imprint of Elsevier.
- [3] Khawandi, S., Daya, B., & Chauvet, P. (2011). Implementation of a Monitoring System for Fall Detection in Elderly Healthcare. *Procedia Computer Science*, 216–220.
- [4] Liu, H., & Motoda, H. (1998). *Feature Extraction, Construction and Selection: A Data Mining Perspective*. Spinger US.
- [5] MacLennan, J., Tang, Z., & Crivat, B. (2009). *Data Mining with Microsoft SQL Server 2008*. India, Indianapolis: Wiley Publishing, Inc.
- [6] MDPI. (2014, July 18). *US National Library of Medicine*. Retrieved from Automatic Fall Monitoring: A Review: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4166886/#b57-sensors-14-12900>
- [7] Ntanasis, P., Pippa, E., Özdemir, A. T., Barshan, B., & egalooikononou, V. (n.d.). Investigasi of sensor placement for accurate fall detection. Retrieved July 5, 2018, from ntanasis@ceid.upatras.gr
- [8] P.Rumeau, Bourke, A., O'Laughlin, G., Lundy, J., & Noury, N. (2008). A proposal for the classification and evaluation of fall detectors. 340–349.
- [9] Rokach, L., & Maimon, O. (2008). *DATA MINING WITH DECISION TREES*. Singapore: World Scientific Publishing.
- [10] Rokach, L., & Maimon, O. (2015). *DATA MINING WITH DECISION TREES Theory and Applications*. Singapore: World Scientific.
- [11] Santosa, B. (2007). *Data mining Teknik Pemanfaatan Data untuk Keperluan Bisnis*. Yogyakarta: GRAHA ILMU.
- [12] Susanto, S., & Suryadi, D. (2010). *Pengantar Data Mining Menggali Pengetahuan dari Bongkahan Data*. Yogyakarta: ANDI OFFSET.