

BAB III METODOLOGI

2.6 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. Beberapa fitur diambil dari sinyal akselerasi, seperti mean, standar deviasi, jumlah besaran *vector*, dan sudut kemiringan sederhana, namun kurang informatif 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 .

2.6.1 Data uji coba

Data uji coba yang digunakan diambil dari 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:

- a. Walking
- b. Falling
- c. Sitting
- d. Lying
- e. Lying Down
- f. Standing Up from Sitting
- g. Standing Up from Lying

Setiap *dataset* memiliki kategori aktifitas dengan nilai fitur masing-masing seperti pada Lampiran A. Kategori aktifitas tersebut diambil dari nilai 20 fitur yang menggambarkan prediksi posisi *activity*. :

- a. min_data_x
- b. max_data_x
- c. rerata_x
- d. std_x
- e. var_x
- f. min_data_y
- g. max_data_y
- h. rerata_y
- i. std_y
- j. var_y
- k. min_data_z
- l. max_data_z
- m. rerata_z
- n. std_z
- o. var_z
- p. std_mag
- q. std_sum_vec_mag
- r. std_sum_vec_hor
- s. delta_sum_vec_mag
- t. rms_sum_vec_mag

Menentukan setiap fitur dari masing-masing data dihitung menggunakan beberapa rumus, rumus-rumus untuk percobaan jatuh yang digunakan terdapat pada Tabel 2.6.1 rumus variabel.

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

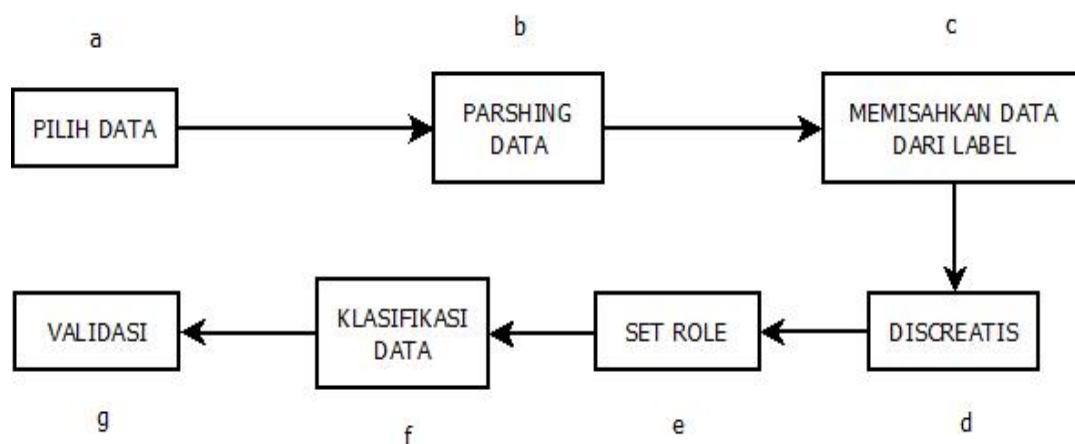
F3	Variance	$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2$
F4	Standard deviation magnitude	$ \sigma = \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2}$
F5	Sum vector magnitude	$ a = \sqrt{a_x^2 + a_y^2 + a_z^2}$
F6	Sum vector on horizontal plane	$ a _h = \sqrt{a_x^2 + a_z^2}$
F7	Standard deviation of 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	$ASMA = \frac{1}{t_2 - t_1} \left(\int_{t_1}^{t_2} \sqrt{a_x^2(t) + a_y^2(t) + a_z^2(t)} dt \right)$

Tabel 2.6.1 rumus variabel

2.7 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 dihitug menggunakan algoritma *decision tree* dan akan memberikan data hasil berupa bagan *decision tree* dan bagan Confusion Matrix. Berikut tahapan-tahapan klasifikasi menggunakan RapidMiner :

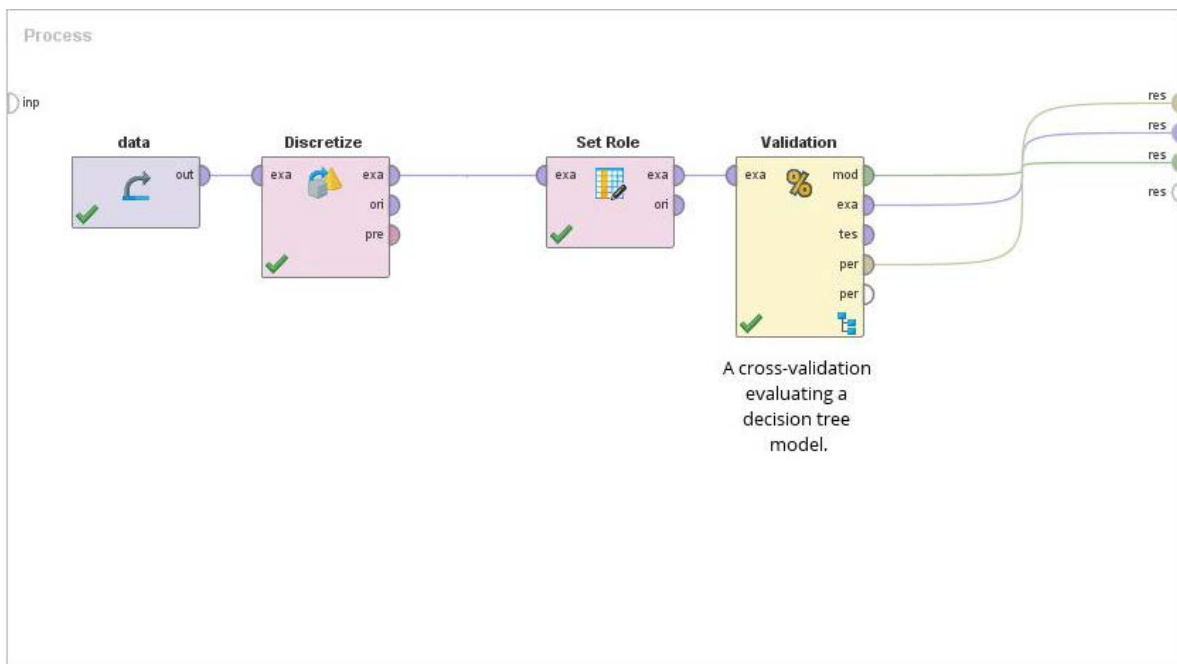


Gambar2.7.1 Proses Klasifikasi

- a. 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.
- b. 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.
- c. 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.

- d. **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.
- e. **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 dimasukkan ke dalam kategori 'label'. Agar pada saat pengkategorian data 'label' tidak ikut serta dihitung dan merubah hasil.
- f. **KLASIFIKASI DATA** : data diklasifikasikan menggunakan algoritma decision tree sesuai dengan spesifikasi yang telah ditentukan.
- g. **VALIDASI** : hasil klasifikasi menggunakan decision tree akan diproses sehingga menghasilkan bagan decision tree, menjadi data pengkategorian sehingga dapat dihitung presentasi keakurasiannya menggunakan confusion matrix.

Sebagai berikut penjelasan lebih detail mulai dari *discretize* hingga validasi :



Gambar 2.7.2 Set operator

Seperti pada Gambar 2.7.2 Set operator setelah data yang ingin digunakan sudah dimasukkan dalam RapidMiner, maka kita sudah dapat menyusun operator-operator dalam RapidMiner untuk mengolah data jatuh dengan menggunakan metode *decision tree*. Agar data dapat dioperasikan dengan maksimal maka berikut operator operator yang digunakan :

1. Retrieve (data) :

Operator ini memuat Objek(data) ke dalam proses. Objek ini sering merupakan *ExampleSet* tetapi juga bisa menjadi model yang disimpan. Didalam operator ini terdapat data dari percobaan jatuh yang telah disiapkan .

2. *Discretize* (by binning) :

Operator ini mendiskritkan atribut numerik yang dipilih ke atribut nominal. Jumlah parameter bins digunakan untuk menentukan jumlah bins yang diperlukan. Diskritisasi ini dilakukan dengan binning sederhana. Kisaran nilai numerik dibagi menjadi satu *segmen* atau kelompok dengan ukuran yang sama. Setiap segmen mewakili bin. Operator ini juga memungkinkan untuk menerapkan binning hanya pada rentang nilai, dapat diaktifkan dengan parameter batasan *define*. Nilai min dan parameter nilai maksimal digunakan untuk menentukan batas kisaran.

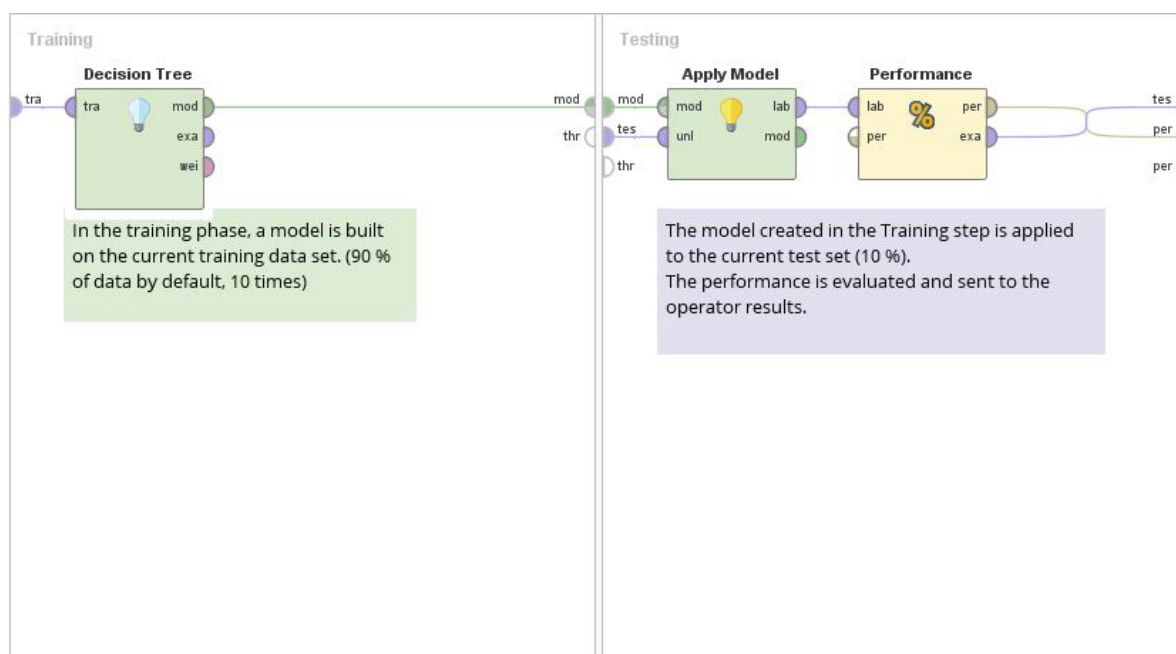
3. *Set Role* :

Operator yang mengklasifikasikan sebuah atribut sebagai atribut khusus atau atribut standar. Contohnya pada suatu data dapat memiliki banyak atribut khusus, tetapi setiap peran khusus hanya dapat muncul satu kali. Jika peran khusus ditugaskan untuk lebih dari satu atribut, semua peran akan diubah menjadi biasa kecuali untuk atribut terakhir.

4. *Validation (Cross Validation)* :

Merupakan operator yang memiliki dua *subproses* : *subproses Training* dan *subproses Testing*. *Subproses Training* digunakan untuk melatih sebuah model. Model yang telah dilatih (diuji atau diproses terlebih dahulu) kemudian diterapkan dalam *subproses Testing*. Kinerja model akan diukur selama *fase Testing*.

Dalam Validation terdapat beberapa operator yang beroperasi terbagi menjadi *training* dan *testing* seperti pada Gambar 2.7.3 Validation.



Gambar 2.7.3 Validation

1. Modeling (*Decision tree*):

Operator Modeling ini dapat memproses set data yang berisi atribut nominal dan numerik. Atribut label harus nominal untuk klasifikasi dan numerik untuk menjalankan proses. Modeling yang digunakan disini adalah modeling *Decision tree*.

2. *Apply Model* :

Operator ini akan menguji set data dari operator lain, yang sering merupakan algoritma pembelajaran. Setelah itu, model ini dapat diterapkan pada data set lainnya. Biasanya bertujuan untuk mendapatkan prediksi pada data yang tidak terlihat atau untuk mengubah data dengan menerapkan *model preprocessing*.

3. *Performance* :

Operator ini dapat digunakan untuk semua jenis tugas belajar. Operator ini secara otomatis menentukan jenis tugas pembelajaran dan menghitung kriteria yang paling umum untuk jenis itu.

Setiap operator dihubungkan sesuai dengan kebutuhan yang kita inginkan mulai dari :

- a. Retrieve/data dihubungkan (out) dengan operator (exa) discretize.
- b. Discretize (exa) dihubungkan dengan operator (exa) Set Role.
- c. Set Role (exa) dihubungkan dengan operator (exa) Validation.

Di dalam operator *Validation* :

- a. *Validation suboperator (tra)* dihubungkan dengan operator *(tra) Decision tree*.
- b. *Decision tree (mod)* dihubungkan dengan *suboperator (mod) training*.
- c. *Validation suboperator (mod)* terhubung dengan operator *(mod) Apply Model* dan *Validation suboperator (tes)* terhubung dengan operator *(unl) Apply Model*.
- d. *Apply Model (lab)* terhubung dengan operator *(lab) performance*.
- e. *Performance (per)* dihubungkan dengan *suboperator (per) testing* dan *Performance (exa)* dihubungkan dengan *suboperator (tes) testing*.

Validation (mod) (exa) (per) dihubungkan dengan *(res) port* akhir untuk mengeluarkan hasil bagan *decision tree, performance, dan dataset*.

2.8 Validasi

Validasi merupakan suatu proses *evaluasi* yang bersifat seksama dan cermat yang dihasilkan suatu prosedur dengan nilai yang dapat diterima. *Validasi* dapat memastikan bahwa prosedur yang tertulis memiliki detail yang cukup jelas, sehingga dapat dilaksanakan atau dilanjutkan oleh analis yang berbeda dengan hasil yang sebanding. *Validasi* digunakan untuk metode tidak baku, metode-metode yang dikembangkan atau metode baku yang telah dimodifikasi. Dilakukannya validasi untuk memastikan bahwa metode pengujian maupun kalibrasi tersebut sesuai untuk penggunaan yang dimaksudkan, dan mampu menghasilkan data yang valid.

Berbagai masalah mutu yang timbul dimana permasalahan-permasalahan tersebut tidak terdeteksi dengan pengerjaan rutin yang dilaksanakan. Adapun maksud dan tujuan sebagai berikut :

- a. Validasi bertujuan untuk memastikan dan mengkonfirmasi bahwa metode analisis tersebut sesuai untuk peruntukannya.
- b. Untuk menghasilkan hasil analisis yang lebih baik. Untuk memaksimalkan hasil analisis tentunya terdapat beberapa parameter yang dinilai dalam validasi, berikut parameter validasi :
 1. Ketepatan (akurasi)

Merupakan ketelitian metode analisis atau ketepatan antara nilai terukur dengan nilai yang diterima, baik nilai konfensi, nilai sebenarnya atau nilai rujukan akurasi diukur sebagai banyaknya analit yang diperoleh kembali pada suatu pengukuran.

2. Presisi

Presisi merupakan kedekatan antara serangkaian hasil analisis yang diperoleh dari beberapa kali pengukuran pada sampel yang sama.

Data set akan diproses menggunakan RapidMiner dengan algoritma *decision tree*. Karena menggunakan algoritma *decision tree* untuk menemukan skema algoritma yang lebih baik, menentukan beberapa percobaan dengan mengubah beberapa parameter pada operator *decision tree*. Parameter pada *decision tree* yang akan digunakan untuk mencari nilai akurasi yang lebih baik *criteria* (*gain ratio*, *gini index*, dan *information gain*), *apply prepruning* (menggunakan *prepruning* atau tidak), *apply pruning* (menggunakan *pruning* atau tidak).

Satu *criteria* menjadi empat skema kemungkinan :

- a. diproses tanpa *pruning* dan *prepruning* sama sekali
- b. diproses dengan menggunakan *pruning* tanpa *prepruning*
- c. diproses menggunakan *prepruning* tanpa menggunakan *pruning*
- d. diproses dengan menggunakan *pruning* dan *prepruning*.

Setiap *criteria* diproses dengan empat skema, maka ada 12 kali percobaan. Dari 12 kali percobaan, skema dengan *criteria* *information gain* diproses dengan menggunakan *pruning* dan *prepruning* memberikan akurasi lebih baik dari pada skema yang lainnya terlihat pada Gambar 4.3.1.3 *Performance*.