

PERANCANGAN DAN IMPLEMENTASI ALGORITMA C4.5 UNTUK DATA MINING ANALISIS TINGKAT RISIKO KEMATIAN *NEONATUM* PADA BAYI

Ifan Junaedi¹, Ndaru Nuswantari², Verdi Yasin³

Departemen Teknik Informatika¹, STMIK Jayakarta¹
Program Studi Teknik Informatika², STMIK Jayakarta²
Departemen Teknik Informatika³, STMIK Jayakarta³
ifan@stmik.jayakarta.ac.id¹, verdiyasin29@gmail.com

ABSTRAK

Kematian *Neonatum* merupakan masalah yang banyak dialami negara berkembang, termasuk Indonesia. Kematian *neonatum* banyak dipengaruhi oleh berbagai faktor, salah satunya adalah faktor sosial biologi pada ibu hamil. Faktor sosial biologi ibu hamil yang akan dibahas dalam penelitian ini meliputi empat variabel, yaitu tempat tinggal, tingkat pendidikan, tingkat pendapatan rumah tangga, dan urutan kelahiran.

Penulis menggunakan data hasil Survei Demografi dan Kesehatan Indonesia tahun 2007 s.d 2012. Data mining dilakukan dengan menggunakan teknik klasifikasi untuk membentuk suatu pohon keputusan. Algoritma C4.5 digunakan untuk menentukan nsiko kematian neonatum per variabel per kategori untuk pohon keputusan yang terbentuk. Algoritma ini dipilih karena C4.5 memiliki berbagai kelebihan dibanding algoritma *Iterative Dichotomiser 3* yang telah dikembangkan sebelumnya.

Program aplikasi dirancang dengan berorientasi objek. Perancangan program dengan menggunakan alat bantu pemodelan *Unified Modeling Language (UML)*. UML yang dipakai yaitu *Use Case Diagram*, *Activity Diagram*, dan *Sequence Diagram*. Penulis menggunakan bahasa pemrograman *Visual Basic. Net* dengan editor *Microsoft Visual Studio*.

Kesimpulan dari penelitian ini adalah aplikasi yang dibangun menggunakan C4.5 dapat mendukung teknik analisis risiko kematian *neonatum*. Pohon keputusan yang dibangun masih terbatas empat level dengan maksimal tiga percabangan untuk masing-masing node. Oleh karena itu, untuk pengembangan aplikasi selanjutnya diharapkan mampu menganalisis lebih dari empat variabel (level).

Kata kunci: *neonatum*, data mining, algoritma C4.5, pohon keputusan.

I. PENDAHULUAN

A. Latar Belakang

Badan Pusat Statistik (BPS) merupakan lembaga pemerintah yang bertugas melakukan kegiatan pengumpulan data statistik. Kegiatan pengumpulan data dilakukan melalui sensus dan survei. Kewenaagan ini diatur dalam Undang-undang Nomor 16 Tahun 1997 tentang Statistik. Salah satu data yang dikumpulkan adalah data Survei Demografi dan Kesehatan Indonesia (SDKI).

Dari kegiatan survei tersebut salah satu data yang diperoleh adalah variabel-variabel untuk mengukur risiko tingkat kematian bayi. Kematian *neonatum* adalah kematian bayi dalam bulan pertama setelah lahir. Risiko kematian bayi dipengaruhi oleh beberapa faktor, salah satunya adalah faktor sosial biologi ibu hamil. Faktor sosial biologi tersebut terdiri dari variabel tempat tinggal, tingkat pendidikan ibu, tingkat pendapatan keluarga, dan urutan kelahiran anak. Tempat tinggal dibedakan menjadi perkotaan dan pedesaan. Tingkat pendidikan ibu dibagi dua, yaitu pendidikan rendah (<

SLTP), dan tinggi (SLTP +). Tingkat pendapatan keluarga digolongkan menjadi dua, yaitu menengah ke atas dan menengah ke bawah. Urutan kelahiran digolongkan menjadi kelahiran pertama (1), kedua (2), dan ketiga atau lebih (3+). Algoritma C4.5 merupakan salah satu algoritma induksi pohon keputusan yang dikembangkan oleh J.Ross Quinlan. Algoritma ini memiliki kelebihan dibandingkan algoritma *Iterative Dichotomiser 3*. Algoritma C4.5 digunakan untuk menganalisis peluang suatu kejadian dengan indikator dari beberapa variabel. Penelitian kali ini akan melakukan analisis dan implementasi algoritma C4.5 untuk data mining mengukur tingkat risiko kematian bayi dari data SDKI.

B. Identifikasi masalah

Kematian *neonatum* adalah kematian bayi dalam bulan pertama setelah lahir. Peluang terjadinya *neonatum* dapat digunakan sebagai indikator yang mencerminkan target kelangsungan hidup anak dan kondisi sosial, ekonomi, dan kesehatan reproduksi. Masalahnya adalah

menentukan alat analisis apa yang dapat digunakan untuk mengukur tingkat risiko terjadinya kematian tersebut.

C. Rumusan masalah

Berdasarkan latar belakang dan identifikasi masalah maka dapat dirumuskan, beberapa hal diantaranya adalah sebagai berikut :

1. Apa saja variabel yang dapat digunakan sebagai indikator pengukur tingkat kematian *Neonatum*.
2. Apakah aplikasi yang dibangun dengan algoritma *C4.5* dapat mendukung teknik data mining.
3. Bagaimana karakteristik pohon keputusan (*decision tree*) dengan algoritma *C4.5* yang dapat dihasilkan oleh aplikasi
4. Bagaimana merancang dan mengimplementasikan algoritma *C4.5* menjadi sebuah Aplikasi / sistem yang dapat membantu pengguna melakukan analisis risiko kematian *neonatum*.

D. Pembatasan masalah

Penelitian berikut ini memiliki ruang lingkup, sebagai berikut :

1. Area permasalahan hanya untuk analisis risiko kematian *neonatum* dengan hanya memperhatikan variabel-variabel tempat tinggal, tingkat pendidikan ibu, tingkat pendapatan keluarga, dan urutan kelahiran anak.
2. Studi kasus yang digunakan adalah data Survei Demografi dan Kesehatan Indonesia, Badan Pusat Statistik, data tahun 2007 s.d 2012
3. Algoritma yang digunakan untuk melakukan analisis data mining adalah algoritma *C4.5*
4. Aplikasi/sistem yang dibangun hanya sebagai alat bantu pendukung keputusan, karena masih mungkin dilanjutkan oleh uji-uji analisis lainnya.

E. Tujuan Penelitian

Penelitian ini bertujuan untuk :

1. Melakukan analisis dan perancangan aplikasi data mining induksi decision tree dengan algoritma *C4.5*
2. Mengimplementasikan teknik klasifikasi untuk data mining risiko kematian *neonatum* dengan studi kasus data Survei Demografi dan Kesehatan Indonesia.

3. Melakukan implementasi algoritma *C4.5* untuk dalam membangun induksi pohon keputusan.
4. Melakukan analisis risiko kematian *neonatum* berdasarkan empat variabel karakteristik sosial ekonomi ibu hamil.
5. Melakukan uji coba terhadap aplikasi data mining yang dibangun dengan membandingkan hasilnya terhadap alat uji analisis yang lainnya.

F. Metodologi Penelitian

Metodologi penelitian yang digunakan dalam penyusunan skripsi ini, meliputi:

1. Studi Pustaka

Studi Pustaka dilakukan untuk mengumpulkan landasan teoritis pengembangan sistem termasuk beberapa konsep dan defmisi yang terkait dengan area permasalahan sistem.

2. Penelitian Sekunder

Penelitian sekunder merupakan teknik melakukan penelitian / analisis berdasarkan data yang telah ada (dikumpulkan oleh pihak lain). Peneliti menggunakan data karakteristik ibu hamil yang berasal dari Survei Demografi dan Kesehatan Indonesia.

II. LANDASAN TEORI

A. Penelitian terkait Kasus *Neonatum* di Indonesia

1. Survei Demografi dan Kesehatan Indonesia

Survei Demografi dan Kesehatan Indonesia (SDKI) merupakan salah satu survei di bidang kependudukan dan kesehatan oleh Badan Pusat Statistik. SDKI dirancang untuk menyajikan informasi mengenai tingkat kelahiran, kematian, keluarga berencana, dan kesehatan. Pola pertanyaan ini sama dengan pola DHS (*Demographic and Health Surveys*) internasional. Salah satu analisis yang dapat dilakukan dari data hasil survei ini adalah studi tentang kematian (*mortalitas*), khususnya potensi kematian bayi pada usia bulan pertama lahir.

Kematian merupakan salah satu masalah yang berpengaruh terhadap struktur penduduk suatu negara. Tingkat kematian suatu bangsa menggambarkan tingkat kesejahteraan suatu negara. Kematian pada usia baru lahir (bayi) merupakan salah satu masalah yang dihadapi bangsa Indonesia. Angka kematian bayi merupakan suatu barometer dari tingkat kesehatan masyarakat. Kematian bayi baik sebelum lahir maupun sesudah lahir merupakan

barometer kesehatan suatu masyarakat dan merupakan suatu keadaan yang perlu diperhatikan karena turut mempengaruhi perubahan penduduk. Berdasarkan data SDKI tahun 2007, kematian anak di bawah satu tahun adalah sebesar 34 per 1000 kelahiran hidup. Angka ini sedikit menurun dibandingkan dengan angka kematian bayi (AKB) tahun 2002 adalah 35 per 1000 kelahiran hidup. Sekitar dua pertiga kematian bayi di bawah umur satu tahun terjadi pada masa *neonatum* (bulan pertama setelah lahir).

2. Kematian Neonatum

Kematian atau *mortalitas* adalah berakhirnya kehidupan dengan ketiadaan nyawa dalam organisme biologis. Kematian adalah peristiwa hilangnya semua tanda-tanda kehidupan secara permanen yang bisa terjadi tiap saat setelah kelahiran hidup. (Budi Utomo, 1985). Jenis kematian bayi ada dua yaitu kematian di dalam rahim dan kematian di luar rahim. Kematian di dalam rahim (*intra uterin*) terdiri atas kematian *abortus*, *immatur*, dan *prematum*. *Abortus* adalah kematian janin menjelang dan samapai 16 minggu. *Immatur* adalah kematian janin antara umur kandungan di atas 16 minggu sampai pada umur kandungan 28 minggu. Sedangkan *prematum* adalah kematian janin di dalam kandungan pada umur di atas 28 minggu sampai dengan kelahiran. Kematian bayi di luar rahim (*extra uterin*) terdiri dari lahir mati, kematian bayi baru lahir, kematian lepas baru lahir, dan kematian bayi. Lahir mati (*still birth*) adalah kematian bayi yang cukup masanya pada waktu keluar dari rahim dan tidak ada tanda-tanda kehidupan. Kematian bayi baru lahir (*neonatum death*) adalah kematian bayi sebelum berumur satu bulan. Kematian lepas baru lahir (*post neonatum death*) adalah kematian bayi setelah berumur satu bulan tetapi kurang dari satu tahun. Kematian bayi (*infant mortality*) adalah kematian setelah bayi lahir hidup hingga berumur kurang dari satu tahun.

Sebagaimana telah disebutkan di atas, kematian *neonatum* merupakan kematian bayi lahir hidup dalam bulan pertama setelah lahir. Angka kematian neonatum dipengaruhi oleh kelahiran hidup. Kelahiran hidup (*live birth*) didefinisikan sebagai keluarnya hasil konsepsi secara sempurna dari ibunya tanpa memandang lamanya kehamilan dan sesudah terpisah dari ibunya dapat bernafas atau menunjukkan tanda-tanda kehidupan. Kematian bayi baru lahir disebabkan berbagai hal, yaitu faktor medis, faktor sosial, biologi, dan kegagalan sistem yang dipengaruhi oleh keadaan dan kultur.

Kesehatan bayi baru lahir berkaitan erat dengan faktor ibu. Kematian ibu, janin, dan *neonatum* di negara berkembang biasanya terjadi karena persalinan terjadi di rumah, tanpa pertolongan tenaga kesehatan, dan terdapat keterlambatan akses untuk menerima perawatan yang sesuai. Selain faktor medis, faktor sosial ibu juga erat kaitannya dalam mempengaruhi risiko neonatum. Faktor sosial biologi atau karakteristik ibu dapat dilihat dari variabel tempat tinggal, tingkat pendidikan ibu, tingkat pendapatan keluarga, dan urutan kelahiran anak. SDKI membagi tempat tinggal menjadi wilayah perkotaan dan pedesaan. Kriteria tingkat pendidikan ibu dipandang tinggi jika ibu minimal tamat SLIP, dan rendah jika ibu tidak pernah sekolah atau hanya tamat SD. Sedangkan tingkat pendapatan keluarga dibedakan menjadi menengah ke atas dan menengah ke bawah. Urutan kelahiran digolongkan menjadi kelahiran pertama (1), kedua (2), dan ketiga atau lebih (3+).

B. Data Mining

1. Pengertian Data Mining

Data mining adalah serangkaian proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang tidak dapat diperoleh secara manual. Menurut Daniel T. Larose dalam bukunya yang berjudul "*Discovering Knowledge in Data*", data mining adalah sebuah proses menemukan atau menggali lebih dalam mengenai pola, hubungan, maupun *trend* dari sekumpulan data menggunakan sebuah teknik matematika maupun statistik.

Data mining adalah sebuah analisis yang luas dari sekumpulan data observasi untuk mencari hubungan, ringkasan dan pola sebagai cara agar data tersebut dapat dipahami maupun bermanfaat bagi pemilik data (*Hand et al.*).

Data mining adalah disiplin ilmu yang membawa secara bersama-sama berbagai macam teknik machine learning, pengenalan pola, statistik, database dan visualisasi untuk menggali informasi dari basis data yang besar. (*Evangelas Simoudis in Cabena et al*)

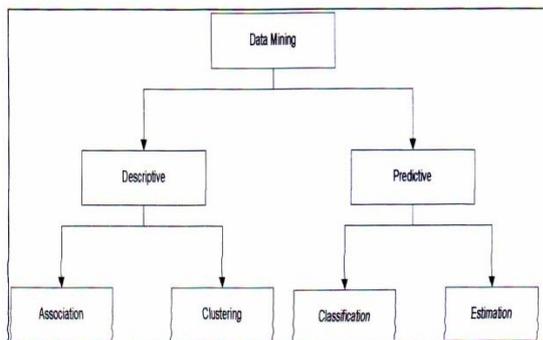
Data mining juga diartikan sebagai perangkat lunak yang digunakan untuk menemukan pola-pola tersembunyi maupun hubungan-hubungan yang terdapat dalam basis data yang besar dan menghasilkan aturan-aturan yang digunakan untuk memperkirakan perilaku di masa mendatang. Data

mining sering dikatakan berkaitan dengan "penemuan pengetahuan" dalam basis data.

Data mining menggunakan berbagai macam teknologi yang diambil dari ilmu statistik, jaringan saraf (*neural network*), logika fuzzy (*fuzzy logic*), algoritma genetika (*genetic algorithm*), dan berbagai teknologi kecerdasan buatan lainnya. Selain itu, pendekatan data mining juga dapat dilakukan dengan visualisasi data. Pada pendekatan ini pengguna dibantu untuk menemukan sendiri pola dari sejumlah data berukuran besar dengan didasarkan visualisasi oleh data mining.

2. Teknik – teknik Data Mining

Tujuan dari data mining adalah untuk mendeskripsikan apa yang telah terjadi (*descriptive data mining*) dan untuk memprediksi apa yang akan terjadi (*predictive data mining*). *Descriptive data mining* dilakukan dengan mencari pola pada kejadian yang telah lampau yang mempengaruhi kejadian yang terjadinya pada masa sekarang. Teknik data mining yang termasuk dalam kategori ini adalah *association* dan *clustering*. *Predictive data mining* menggunakan kejadian yang telah lampau untuk memprediksi apa yang terjadi pada masa yang akan datang. Yang termasuk dalam *predictive data mining* adalah *classification* dan *estimation*. Pengkategorian teknik data mining dapat dilihat pada bagan di bawah ini.



Gambar 2.1 Teknik-teknik data mining

a. Association Rule Mining

Association rule mining merupakan teknik untuk menemukan aturan asosiasi antara suatu kombinasi item. Aturan asosiatif diukur dari dua parameter, yaitu *support* dan *confidence*. *Support* adalah persentase kombinasi item dalam data, sedangkan *confidence* merupakan kuatnya hubungan antar item dalam aturan asosiatif.

Algoritma yang sering digunakan adalah apriori dengan *generate* dan *test*, yaitu pembuatan kandidat kombinasi item yang mungkin berdasar

aturan tertentu lalu dilakukan pengujian apakah kombinasi item memenuhi syarat support minimum. Kombinasi item yang memenuhi syarat tersebut disebut *frequent itemset*. *Frequent itemset* ini nantinya digunakan untuk membuat aturan-aturan yang memenuhi syarat *confidence minimum*.

b. Clustering

Teknik kluster melakukan pengelompokan set data tanpa berdasarkan kelas data tertentu. *Clustering* dipakai juga untuk memberikan label pada kelas data yang belum diketahui. Oleh karena itu *clustering* digolongkan sebagai metode *unsupervised learning*. Prinsip dari *clustering* adalah memaksimalkan kesamaan antar anggota satu kelas dan meminimalkan kesamaan antar kelas. *Clustering* dapat dilakukan pada data yang memiliki beberapa atribut yang dipetakan sebagai ruang multidimensi. Metode yang diperlukan untuk melakukan teknik kluster diantaranya adalah fungsi jarak, normalisasi, dan metode partisi.

c. Classification

Klasifikasi (*classification*) merupakan proses untuk merumuskan model atau fungsi yang menjelaskan atau membedakan konsep maupun kelas data. Tujuannya adalah untuk dapat memperkirakan kelas dari suatu objek yang labelnya tidak diketahui. Data input untuk klasifikasi adalah kumpulan record dimana setiap record disebut *instance* atau sampel. Sebuah *tuple* (x, y) menyatakan pemetaan record, dimana x adalah himpunan atribut dan y adalah label kelas (kategori atau atribut target). Fungsi target / memetakan setiap himpunan atribut x ke salah satu kategori y yang telah didefinisikan.



Gambar 2.2.

Klasifikasi sebagai pemetaan sebuah himpunan atribut input x ke dalam label kategori kelasnya

Beberapa teknik klasifikasi yang digunakan adalah *decision tree classifier*, *rule-based classifier*, *neural network*, *support vector machine* dan *naive Bayes classifier*. Setiap teknik menggunakan algoritma learning untuk mengidentifikasi model yang memberikan hubungan yang paling sesuai

antara himpunan atribut dan label kelas dari data input. Metode klasifikasi yang paling populer adalah *decision tree*. *Decision tree* (pohon keputusan) merupakan metode yang mudah diinterpretasi oleh manusia. Salah satu algoritma *decision tree* adalah algoritma C4.5.

Proses klasifikasi itu sendiri biasanya dibagi menjadi dua fase, yaitu *learning* dan *test*. Pada fase *learning*, sebagian data yang telah diketahui kelas datanya dimasukkan untuk membentuk model estimasi. Pada fase *test*, model yang sudah terbentuk diuji dengan sebagian data lainnya untuk mengetahui akurasi dari model tersebut. Bila akurasinya memenuhi, maka model dapat diterima untuk melakukan prediksi kelas data yang belum diketahui.

d. *Estimation*

Data mining terletak pada berbagai macam disiplin ilmu, salah satu nya adalah statistik. Teknik data mining menggunakan disiplin ilmu ini untuk melakukan estimasi berdasarkan set data masa lalu untuk memprediksi nilai yang mungkin terjadi di masa depan. Statistik inferensia merupakan cabang ilmu statistik yang mampu menjelaskan/mengestimasi suatu nilai berdasarkan sekumpulan training data set. Metode estimasi ada dua yaitu *point estimation* dan *interval estimation*. *Point estimation* menangani data untuk satu waktu tertentu saja, sedangkan *interval estimation* memprediksi suatu data berdasarkan sekumpulan data pada selang waktu tertentu (*time series analysis*). Dikenal juga metode analisis regresi, korelasi, dan trend data sebagai alat untuk melakukan analisis inferensia.

3. Pohon Keputusan (Decision Trees)

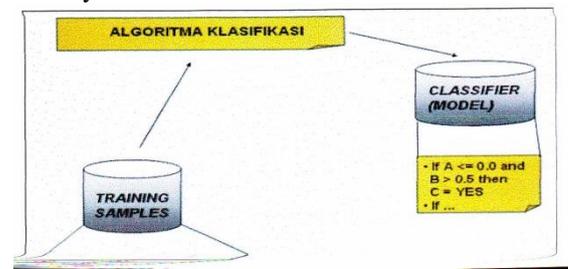
Sesuai uraian di atas, pohon keputusan (*decision tree*) merupakan salah satu metode data mining yang diterapkan dalam teknik klasifikasi. Klasifikasi merupakan proses pembelajaran (*learning*) yang memetakan komponen data ke dalam sejumlah kelas yang telah didefinisikan. Proses klasifikasi menggunakan pendekatan induksi dari sejumlah data sampel yang terdiri dari sejumlah vektor atribut dan sebuah atribut kelas. Tujuan dari proses *learning* adalah untuk memperoleh model klasifikasi yang dikenal dengan sebutan klasifier yang akan memprediksi kelas untuk sebuah sampel berdasarkan nilai dari atribut-atributnya. Proses klasifikasi terdiri dari dua tahap, yaitu:

a. Pembuatan model

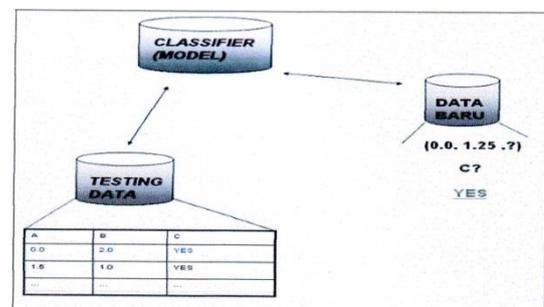
Setiap data diasumsikan telah digolongkan ke dalam sejumlah kelas (*predefined class*). Himpunan data yang akan menyusun model ini disebut *training data*. Model yang dihasilkan direpresentasikan dalam bentuk aturan klasifikasi, pohon keputusan, atau formula matematika.

b. Pemanfaatan model

Tahap ini digunakan untuk mengklasifikasikan obyek yang belum diketahui kelasnya. Estimasi akurasi dilakukan dengan membandingkan kelas dari testing data dengan kelas hasil klasifikasi model. Tingkat akurasi adalah ratio jumlah testing data yang diklasifikasikan secara benar berdasarkan model klasifikasi dengan seluruh jumlah testing data. Jika akurasi diterima maka model klasifikasi dapat digunakan untuk mengklasifikasikan data yang belum diketahui kelasnya.



Gambar 2.3. Ilustrasi Pembuatan Model



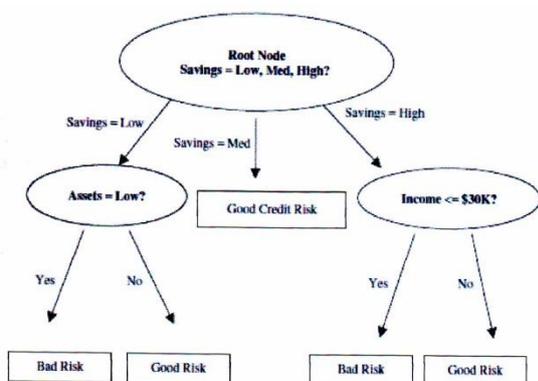
Gambar 2.4. Ilustrasi Pemanfaatan Pohon Keputusan

Pohon keputusan merupakan sistem yang dikembangkan untuk membantu mencari dan membuat keputusan untuk masalah yang berkaitan dengan berbagai macam faktor atau variabel yang berada dalam lingkup masalah tersebut. Melalui pohon keputusan, kita dapat dengan mudah melihat dan mengidentifikasi hubungan antara faktor-faktor yang mempengaruhi suatu masalah untuk kemudian dapat mencari penyelesaian terbaik dengan memperhitungkan faktor-faktor tersebut. Pohon

keputusan ini juga dapat menganalisis nilai risiko dan nilai suatu informasi yang terdapat dalam suatu alternatif pemecahan masalah. Jadi, peran pohon keputusan adalah sebagai alat bantu dalam mengambil keputusan (*decision support tool*).

Dalam melakukan analisis pemecahan masalah, pohon keputusan memetakan mengenai alternatif-alternatif pemecahan masalah yang dapat diambil dari masalah tersebut. Pohon memperhatikan faktor-faktor kemungkinan/peluang yang akan mempengaruhi alternatif-alternatif keputusan tersebut, disertai dengan estimasi hasil akhir yang akan didapat bila alternatif keputusan tersebut dipilih.

Pohon keputusan merupakan model prediksi menggunakan struktur pohon atau struktur berhirarki. Konsep dari struktur pohon keputusan adalah mengubah data menjadi pohon keputusan dan aturan-aturan keputusan. Pohon membagi (*breakdown*) proses pengambilan keputusan yang kompleks menjadi lebih simpel sehingga pengambilan keputusan akan lebih menginterpretasikan solusi dari permasalahan. Pohon keputusan berguna untuk mengeksplorasi data dan menemukan hubungan tersembunyi antara sejumlah calon variabel input dengan sebuah variabel target. Pohon keputusan memadukan antara eksplorasi data dan pemodelan. Contoh dari pohon keputusan dapat dilihat dari gambar di bawah ini.



Gambar 2.5.

Model Pohon keputusan *Daniel T. Larose*

C. Algoritma C4.5

1. Pengertian Algoritma C4.5

Algoritma *C4.5* merupakan pengembangan dari algoritma *ID3* (*Iterative Dichotomiser 3*) yang dikembangkan oleh J. Ross Quinlan. Idenya adalah membuat pohon dengan percabangan awal merupakan atribut yang paling signifikan, dan

seterusnya dibagi menjadi percabangan-percabangan sampai aturan terpenuhi. Pohon keputusan itu sendiri diartikan sebagai suatu cara untuk membagi sekumpulan data menjadi himpunan-himpunan yang lebih kecil dengan menerapkan serangkaian rule atau aturan keputusan. Algoritma *IDS* ini pada dasarnya hanya mengulang langkah pemartisian sampai diperoleh keadaan dimana semua sampel pada sebuah simpul tergolong ke dalam kelas yang sama. Setiap jalur dari akar menuju daun akan merepresentasikan aturan keputusan (*decision rule*) yang akan digunakan sebagai prediktor kelas data berikutnya.

Pohon dibangun dengan cara membagi data secara rekursif sehingga 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. Algoritma *C4.5* dapat menangani data numerik baik yang bersifat diskrit maupun kontinyu. Jika suatu set data mempunyai beberapa nilai pengamatan missing value, jika jumlah pengamatan terbatas maka atribut missing value ini dapat diganti dengan nilai rata-rata dari variabel yang bersangkutan.

Pada algoritma ini, pemilihan atribut mana yang akan menempati suatu simpul dilakukan dengan melakukan perhitungan entropi informasi (*information entropy*) dan mencari nilai yang paling minimum. Pemilihan atribut pada algoritma ini berdasarkan asumsi bahwa kompleksitas yang dimiliki oleh pohon keputusan sangat berkaitan erat dengan jumlah informasi yang diberikan oleh nilai-nilai atributnya. Dengan kata lain, atribut dipilih berdasarkan perolehan informasi terbesar (*highest information gain*) dalam menghasilkan subpohon (*subtree*).

Beberapa kriteria yang dimiliki oleh algoritma induktif *C4.5* adalah:

a. Attribute-value description

Himpunan data yang digunakan untuk menganalisis harus dapat direpresentasikan dalam bentuk himpunan atribut. Tiap atribut dapat memiliki nilai diskrit maupun kontinyu.

b. Predefined classes

Kategori yang akan diberikan kepada tiap sampel harus ditentukan terlebih dahulu.

c. Kelas diskrit, Sebuah kasus atau sampel harus tergolong atau tidak tergolong ke dalam sebuah kelas tertentu dan jumlah sampel harus jauh lebih besar daripada jumlah kelas yang ada.

d. Jumlah data yang mencukupi

Jumlah data yang dibutuhkan dipengaruhi oleh jumlah atribut dan kelas serta kompleksitas dari model klasifikasi yang digunakan.

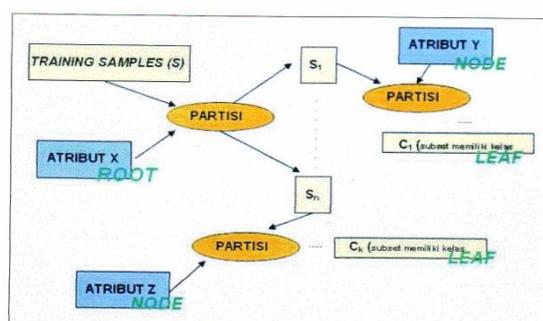
e. Model klasifikasi logis

Pendekatan induktif digunakan untuk membangun classifier yang dapat diekspresikan sebagai pohon keputusan atau aturan keputusan.

Misal training samples T , atribut $(A_1, A_2, A_3, A_4, \dots)$ dan kelas terdiri dari $(K_1, K_2, K_3, K_4, \dots)$, maka kerangka utama dari algoritma C4.5 dapat diuraikan sebagai berikut :

- 1) Jika T tidak kosong dan semua sampel yang ada di dalamnya memiliki kelas K_1 yang sama maka pohon keputusan untuk T adalah sebuah simpul daun (*leaf node*) dengan label K_1
- 2) Jika atribut kosong, maka pohon keputusan berisi sebuah simpul daun dengan label K_j , dimana K_j adalah kelas yang paling dominan pada training samples T
- 3) Jika T terdiri dari sampel yang memiliki kelas yang berbeda-beda maka T dipartisi ke dalam $T_1, T_2, T_3, \dots, T_n$, training samples T dipartisi berdasarkan *distinct value* dari atribut A_k yang pada saat itu menjadi *node parent*. Misalkan A_k terdiri dari 4 jenis nilai n_1, n_2, n_3, n_4 maka T akan dipartisi ke dalam 3 subset yaitu nilai $A_k = n_1, A_k = n_2, A_k = n_3$, dan $A_k = n_4$.

Proses ini dilakukan terus seara rekursif dengan *base case* langkah 1 dan 2. Cara untuk mencari atribut yang akan menjadi *node parent* pada suatu iterasi dilakukan dengan menghitung sebuah kriteria yang disebut gain. Gain berfungsi untuk memilih atribut yang akan diuji berdasarkan konsep teori informasi entropi. Di bawah ini merupakan gambar proses algoritma C4.5.



Gambar 2.6 Proses Algoritma C4.5

2. Komponen Penyusun Algoritma C4.5

Berikut ini akan dijelaskan komponen - komponen yang menyusun algoritma C4.5 dalam membentuk pohon keputusan :

a. Entropi

Entropi merupakan distribusi probabilitas dalam teori informasi dan diadopsi ke dalam algoritma C4.5 untuk mengukur tingkat homogenitas distribusi kelas dari sebuah set data. Semakin tinggi tingkat entropi dari sebuah data set maka semakin homogen distribusi kelas pada data set tersebut. Jika distribusi probabilitas dari kelas didefinisikan dengan $P = (P_1, P_2, P_3, \dots, P_k)$ maka entropi dapat dituliskan sebagai persamaan dari:

$$E(p_1, p_2, \dots, p_k) = - \sum_{i=1}^k (p_i * \log_2(p_i)) \dots (2.1)$$

Persamaan 2. 1 sama dengan persamaan $Info(T)$ sebagai berikut :

$$Info(T) = - \sum_{i=1}^k \left(\frac{frequency(C_i, T)}{|T|} \right) * \log_2 \left(\frac{frequency(C_i, T)}{|T|} \right) \dots (2.2)$$

dimana frekuensi(C_i, T) adalah jumlah sampel di himpunan T yang memiliki kelas $C_1, C_2, C_3, \dots, C_k$.

Sebagai contoh, distribusi kelas (0.5, 0.5) lebih homogen bila dibandingkan dengan distribusi (0.67, 0.33) sehingga distribusi (0.5, 0.5) memiliki entropi yang lebih tinggi dari distribusi (0.67, 0.33). Hal ini dapat dibuktikan sebagai berikut:

$$E(0.5, 0.5) = -0.5x \log_2(0.5) - 0.5x \log_2(0.5) = 1$$

$$E(0.67, 0.33) = -0.67x \log_2(0.67) - 0.33x \log_2(0.33) = 0.91$$

Setelah T dipartisi ke dalam sejumlah subset $T_1, T_2, T_3, \dots, T_n$ berdasarkan atribut X maka perhitungan Info dilakukan dengan menggunakan himpunan training data yang merupakan hasil partisi sebagai berikut:

$$Info_x(T) = \sum_{i=1}^n \left(\frac{T_i}{T} \right) * Info(T_i) \dots (2.3)$$

b. Information Gain

Setelah membagi data set berdasarkan sebuah atribut ke dalam subset yang lebih kecil, entropi dari data tersebut akan berubah. Perubahan entropi ini dapat digunakan untuk menentukan bagus tidaknya pembagian data yang telah dilakukan. Perubahan entropi ini disebut dengan *information gain*. *Information gain* diukur dengan menghitung selisih antara entropi data set sebelum dan sesudah pembagian (*splitting*) dilakukan. Pembagian yang terbaik akan menghasilkan entropi subset yang paling kecil, dengan *information gain* yang terbesar.

Jika sebuah data set D dipartisi berdasarkan nilai dari atribut X dan menghasilkan subset (T_1, T_2, \dots, T_n) maka *information gain* dapat dihitung dengan persamaan :

$$Gain(x) = Info(T) - Info_x(T) \dots (2.4)$$

$Info(T)$ adalah entropi dari data set sebelum dipartisi berdasarkan atribut X , dan $Info_x(T)$ adalah Info dari subset setelah dilakukan pemartisian berdasarkan atribut X .

c. Gain Ratio

Gain ratio merupakan normalisasi dari *information gain* yang memperhitungkan entropi dari distribusi probabilitas subset setelah dilakukan proses partisi. Secara matematis, *gain ratio* dihitung sebagai berikut:

$$GainRatio(X) = \frac{Gain(x)}{Splitinf o(x)} \dots\dots (2.5)$$

dimana :

$$SplitInfo(x) = - \sum_{i=1}^n \left(\left| \frac{T_i}{T} \right| \right) * \log_2 \left(\left| \frac{T_i}{T} \right| \right) \dots$$

(2.6)

$|T_i|$ adalah kardinalitas dari subset T_i yang berada dalam training data T .

D. Object Oriented Programming

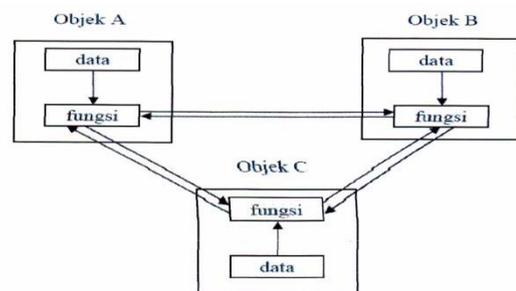
Program adalah rangkaian kegiatan atau perintah untuk dieksekusi oleh komputer. Program dapat berupa utilitas atau aplikasi untuk mencapai tujuan komputasi (pengolahan) tertentu. Algoritma merupakan rangkaian alur berpikir yang diperlukan untuk menulis rangkaian instruksi dalam program. Algoritma adalah rancangan penyelesaian masalah dengan prosedur dan terstruktur, rangkaian perintah dan langkah tatacara yang teratur. Struktur dasar algoritma terdiri dari runtunan (*sequence*), pemilihan (*selection*), dan pengulangan (*repetition*).

Pemrograman berorientasi objek atau *Object Oriented Programming* (OOP) adalah cara pemrograman yang mendasarkan segala sesuatunya pada objek. OOP adalah suatu cara baru dalam berpikir secara berlogika dalam menghadapi masalah-masalah yang akan diatasi dengan bantuan komputer (Nugroho, 2004). Pendekatan OOP melihat permasalahan lewat pengamatan dunia nyata dimana setiap objek adalah entitas tunggal yang memiliki kombinasi struktur data dan fiangsi tertentu. Hal ini berbeda dengan pendekatan terstruktur dimana struktur data dan fungsi didefinisikan secara terpisah dan tidak berhubungan dengan erat.

Pendekatan berorientasi objek membuat data terbungkus pada setiap fungsi atau prosedur dan melindunginya terhadap perubahan yang tidak dikehendaki dari fungsi yang berada di luar.

Pendekatan OOP memiliki karakteristik sebagai berikut :

1. Pendekatan lebih pada data dan bukannya pada prosedur atau fungsi
2. Program besar dibagi pada objek-objek
3. Struktur data dirancang dan menjadi karakteristik dari objek-objek
4. Fungsi-fungsi yang mengoperasikan data tergabung dalam objek yang sama
5. Data tersembunyi dan terlindung dari fungsi/prosedur yang ada di luar.
6. Objek-objek dapat saling berkomunikasi dengan saling mengirim pesan satu sama lain
7. Pendekatan adalah dari bawah ke atas (*bottom up approach*)

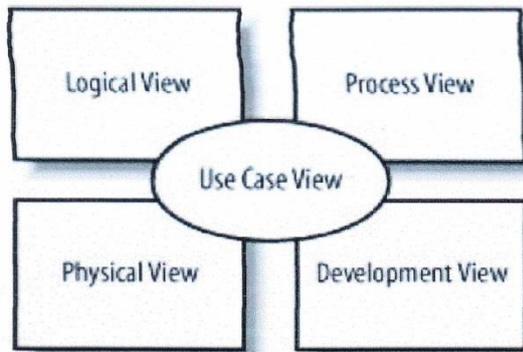


Gambar 2.8. Hubungan Data dan Fungsi Pada Pemrograman Berorientasi Objek

E. Unified Modeling Language

Unified Modeling Language atau UML merupakan peralatan pendukung pemodelan visual dan perancangan program pada OOP. UML merupakan turunan dari beberapa metode yang mempunyai kumpulan diagram grafis sebagai kombinasi dari konsep pemodelan data (*entity relationship diagram*), pemodelan bisnis (*work flow*), pemodelan objek, dan pemodelan komponen (Shoemaker, 2004). Diagram grafis tersebut merupakan tampilan dari beberapa level abstraksi yang dapat digunakan secara bersama oleh semua proses pada seluruh siklus hidup pengembangan perangkat lunak.

UML dibangun di atas 4+1 view. Setiap view berhubungan dengan perspektif tertentu dimana sistem akan diuji. View yang berbeda akan menekankan pada aspek ayng berbeda dari sistem yang mewakili stakeholder tertentu. Model 4+1 tersebut diantaranya use case, logical view, process view, physical view dan development view (Hamilton, 2006). Gambar di bawah ini menggambarkan model 4+1 view UML.



Gambar 2.9. Model 4 + 1 View UML

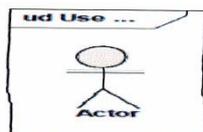
Use case view mendefinisikan perilaku eksternal sistem, seperti *end user*, analis, dan tester. *Logical view* mendeskripsikan struktur logika yang mendukung fungsi-fungsi yang dibutuhkan *use case*. *Process view* mendeskripsikan proses-proses yang terjadi dalam sistem. *Physical view* menjelaskan komponen-komponen fisik dari sistem yang akan dibangun. Sedangkan *development view* menjelaskan bagaimana komponen-komponen fisik didistribusikan ke lingkungan fisik dimana sistem akan dijalankan.

Use Case Diagram

Use Case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. Cara kerja *use case* adalah mendeskripsikan tipikal interaksi antara user (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai (Munawar, 2005). Diagram *use case* menyajikan interaksi antara *use case* dan *actor*. Diagram ini mendeskripsikan persyaratan fungsional sistem serta kebiasaan *actor* berinteraksi pada batasan sistem. Komponen-komponen dalam diagram *use case* adalah :

a. Aktor

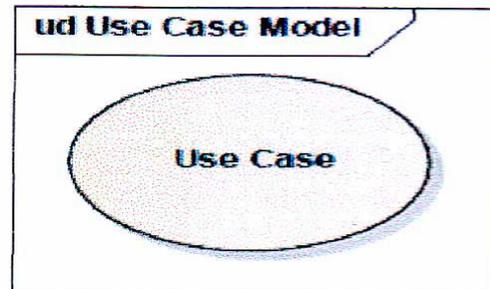
Aktor adalah pengguna sistem, dapat berupa manusia, mesin, dan sistem lainnya. Semua yang berinteraksi dengan sistem dari luar atau batasan sistem disebut dengan aktor. Representasi dari aktor dalam UML adalah:



Gambar 2.10. Gambar use case representasi actor

b. Use case

Use case adalah elemen dalam pemodelan yang menggambarkan bagaimana pengguna dalam sistem akan berinteraksi dengan sistem untuk melakukan suatu pekerjaan. Notasi *use case* dalam UML direpresentasikan dengan gambar 2.11.

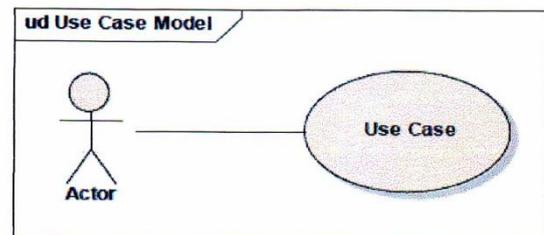


Gambar 2.11. Gambar use case representasi *use case*

c. Relasi

Relasi dalam *use case* diagram secara umum terdiri dari:

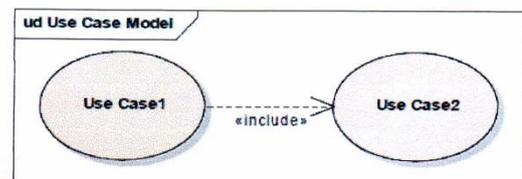
1) Relasi asosiasi, menggambarkan relasi antara aktor dengan *use case*. Relasi asosiasi dalam UML direpresentasikan dalam garis.



Gambar 2.12. Representasi relasi asosiasi

2) Relasi *antar-use case*, terdiri dari relasi *include* dan *extend*.

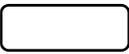
Relasi *include* memungkinkan satu *use case* menggunakan fungsionalitas yang disediakan oleh *use case* lainnya. Relasi *extend* memungkinkan satu *use case* secara opsional menggunakan fungsionalitas yang disediakan oleh *use case* lainnya.



Gambar 2.13. Representasi relasi *include*

2. Activity Diagram

Activity diagram adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis, dan aliran kerja. Activity diagram digunakan untuk memodelkan perilaku sebuah sistem. Beberapa komponen dalam activity diagram disajikan dalam tabel 2.2. Beberapa Komponen dalam Activity Diagram

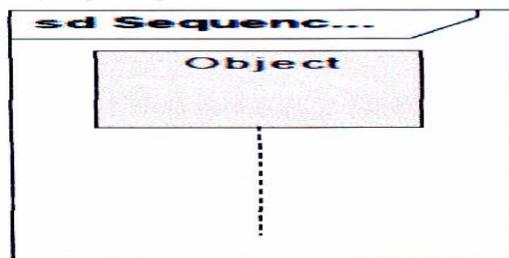
Simbol	Keterangan
•	Titik awal
●	Titik akhir
	Activity
	Action
	Pilihan untuk pengambilan keputusan
	
	Aliran akhir (flow final)

Tabel 2.2 Beberapa Komponen dalam Activity Diagram

3. Sequence Diagram

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh object dan message (pesan) yang diletakkan di antara object, message, dan waktu. a) Object/participant

a) Object/participant



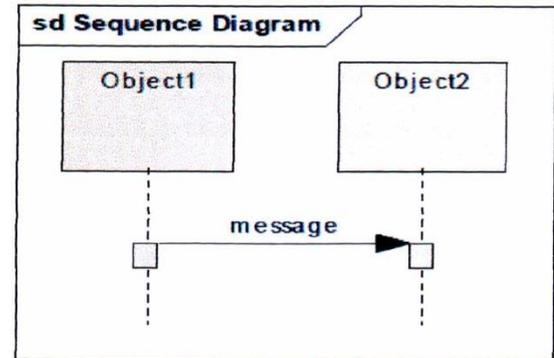
Gambar 2.15. Representasi object/participant

Participant diletakkan di bagian atas diagram dari kiri ke kanan. Setiap participant terhubung dengan garis titik-titik yang disebut lifeline. Sepanjang lifeline terdapat kotak yang disebut activation,

yang mewakili sebuah eksekusi operasi dari participant

b) Message

Message menunjukkan sebuah aliran informasi atau perpindahan kontrol di antara dua participant. Sebuah message bergerak dari satu participant ke participant lain dan dari satu lifeline. 1



Gambar 2.16. Representasi message

c) Time

Time adalah diagram yang mewakili waktu pada arah vertikal. Waktu dimulai dari atas ke bawah. Message yang berada di atas akan dijalankan terlebih dahulu daripada message di bawahnya.

III. ANALISIS DAN PERANCANGAN

A. Tinjauan Kasus

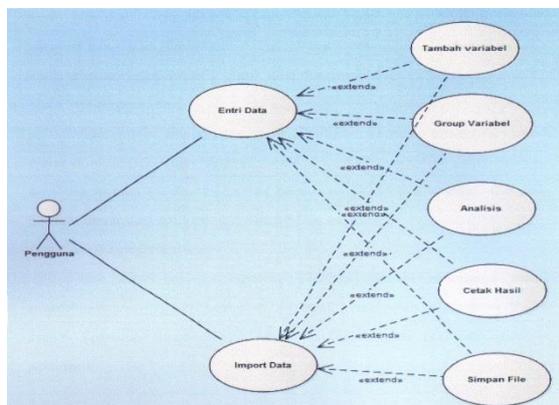
Pemeriksaan kasus risiko kematian monatum dengan memperhatikan empat variabel karakteristik ibu melibatkan pemeriksaan kesesuaian training data set yang dipakai. Sebelum melakukan pembentukan tree dengan algoritma C4.5, training data set yang dipakai diperiksa apakah jumlah variabel yang telah di-input lebih dari satu. Dalam analisis ini, jumlah variabel yang digunakan adalah hanya satu class variable dan minimal satu attribute variable. Algoritma C4.5 menginduksi pembentukan tree (pohon) dengan berdasarkan nilai entropi dan information gain dari setiap kategori dalam variabel. Pada algoritma ini, pemilihan atribut mana yang akan menempati suatu simpul dilakukan dengan melakukan perhitungan entropi (information entropy) dengan mencari nilai yang paling minimum. Pemilihan atribut pada algoritma ini berdasarkan pada asumsi bahwa kompleksitas yang dimiliki oleh pohon keputusan sangat berkaitan erat dengan jumlah informasi yang diberikan oleh nilai-nilai atributnya. Teknik ini memilih atribut yang memberikan perolehan informasi terbesar (highest information gain) dalam menghasilkan subpohon (subtree) untuk mengklasifikasikan sampel.

Hasil induksi pohon untuk data besar pendapatan bertipe kontinyu dengan algoritma C4.5

C. Spesifikasi Rancangan Program

1. Use Case Diagram

Use case diagram bertujuan menggambarkan fitur-fitur yang terdapat pada program aplikasi yang dibuat dan interaksi yang terjadi antara pengguna dan aplikasi tersebut. *Use case diagram* pada perancangan program aplikasi dalam penelitian ini terdiri dari satu *actor* dan tujuh *use case* (Gambar 3.1). *Actor* yang berinteraksi dengan program aplikasi adalah pengguna.



Gambar 3.4. Rancangan *User Case Diagram*

a. Use Case Entri Data

Use Case ini memberi fasilitas kepada pengguna untuk melakukan entri data yang akan dipakai sebagai learning data set. Pengguna dapat memasukan data pada form *spreadsheet* yang telah disediakan oleh aplikasi. Secara *default*, pada form akan diberikan satu variabel. Baris akan bertambah secara otomatis, sehingga pengguna dapat memasukkan data berapa pun jumlah data nya. Sedangkan untuk jumlah variabel (kolom) tidak bisa dilakukan otomatis. Pengguna harus menentukan terlebih dahulu variabel yang akan ditambahkan. Hal ini untuk menghindari adanya variabel yang sebenarnya tidak perlu untuk dilakukan analisis.

b. Use Case Import Data

Use case ini memberikan fasilitas kepada pengguna untuk melakukan import data dari eksternal file, baik teks file (.txt) maupun database Access (.mdb). Pengguna juga dapat membuka kembali file yang pernah dibuat dari aplikasi (Quinlan file atau .qln).

c. Use Case Tambah Variabel

Use case ini memberikan fasilitas kepada pengguna untuk menambahkan variabel sejumlah variabel yang akan digunakan dalam proses *learning*. Penambahan bisa dilakukan satu per satu maupun sekaligus.

d. Use Case Group Variabel

Pengguna diberikan keleluasaan dalam menentukan variabel mana yang akan menjadi kelas maupun atribut. Variabel kelas sebagai variabel *dependen* (tidak bebas) dan variabel atribut sebagai variabel *independen* (bebas). Pengguna juga dapat menentukan tipe variabel dan pengkodean variabel.

e. Use Case Analisis

Setelah mendefinisikan variabel yang digunakan dan learning data set telah lengkap, pengguna dapat melakukan analisis data. Analisis data merupakan proses pembentukan *tree* dengan algoritma C4.5.

f. Use Case Simpan File

Use case Simpan File memungkinkan pengguna untuk menyimpan pekerjaannya dalam format text file (.qln). File .qln hanya bisa dibaca dengan aplikasi ini. Pengguna dapat membuka kembali file tersebut ketika pengguna akan melakukan tindakan lanjut terhadap data yang ada.

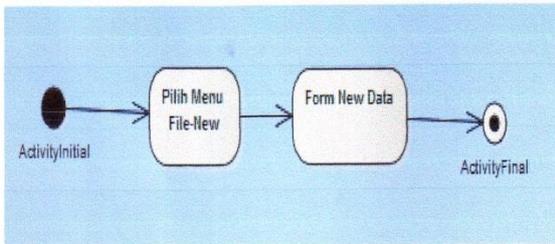
g. Use Case Cetak Hasil

Setelah proses analisis dilakukan, dan dihasilkan suatu bentuk *tree*, kebutuhan selanjutnya adalah mencetak tampilan tersebut. *Use case* cetak hasil memungkinkan pengguna mencetak pohon keputusan yang dihasilkan.

2. Activity Diagram

Activity diagram berfungsi untuk mendeskripsikan logika prosedural, proses bisnis, dan aliran kerja. Sesuai dengan fungsinya *activity diagram* berfungsi untuk menjelaskan secara lebih lanjut tentang *use case* dalam perancangan program aplikasi ini. *Activity diagram* yang digunakan dalam perancangan program aplikasi ini dapat dijelaskan sebagai berikut.

a. Activity diagram untuk Entri Data

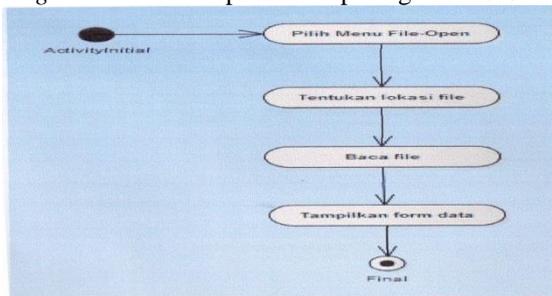


Gambar 3.5. Rancangan *activity diagram* untuk use case Entri Data

Activity diagram menggambarkan alur prosedural pembuatan data/project baru. Dimulai dengan memilih menu File-New dan kemudian akan tampil form entri (*spreadsheet*) kosong yang digunakan untuk menampung data yang akan dimasukkan.

b. *Activity diagram* untuk Import Data

Alur prosedural untuk membuka file diawali dengan memilih menu File-Open. Langkah selanjutnya adalah menentukan lokasi file yang akan dibuka, membaca file yang dipilih dan menampilkan form yang menampung data file tersebut. *Activity diagram* buka file dapat dilihat pada gambar 3.6.

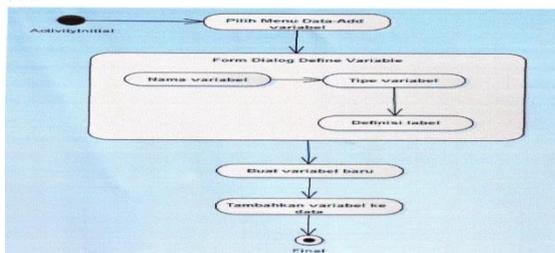


Gambar 3.6.

Rancangan *activity diagram* untuk use case Import Data

c. *Activity diagram* untuk Tambah Variabel

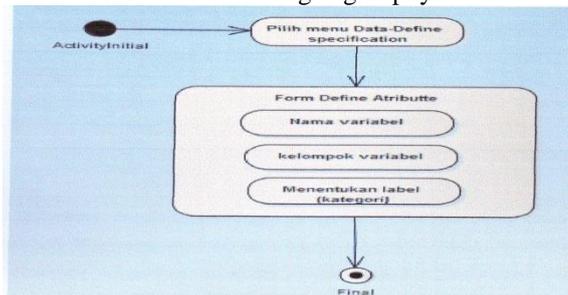
Activity diagram tambah variabel baru menjelaskan prosedur untuk menambahkan sejumlah variabel yang akan digunakan sebagai training data set.



Gambar 3.7. Rancangan *activity diagram* untuk use case tambah variabel

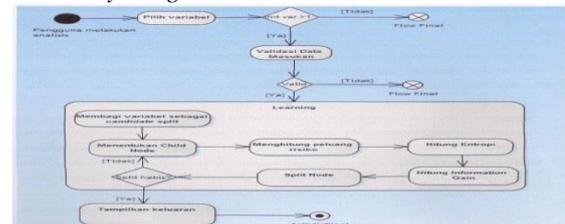
d. *Activity diagram* Group Variabel

Activity diagram in menjelaskan prosedur untuk mendefinisikan variabel-variabel yang telah ditambahkan. Dimulai dengan memilih menu Data-Define specification, kemudian akan muncul form dialog Define Attribute yang berisi daftar variabel. Prosedur selanjutnya adalah memasukkan variabel sesuai groupnya, yaitu variabel kelas dan variabel atribut. Selanjutnya, sistem akan mengenali variabel-variabel tersebut sesuai dengan groupnya.



Gambar 3.8. Rancangan *activity diagram* untuk Use Case group variabel

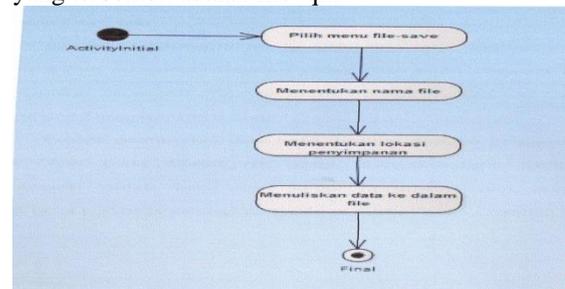
e. *Activity Diagram* Analisis



Gambar 3.9. Rancangan *activity diagram* untuk Use Case Analisis

f. *Activity Diagram* Simpan File

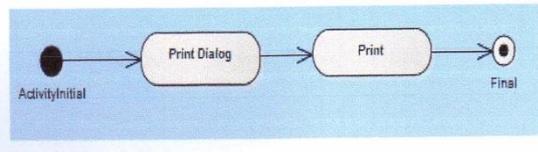
Activity Simpan File menggambarkan proses penyimpanan *learning data set*. Dimulai dengan memilih menu File-Save, kemudian menentukan nama file dan lokasi penyimpanan. Aplikasi akan menuliskan data ke dalam file. File yang terbentuk adalah file .qln.



Gambar 3.10. Rancangan *activity diagram* untuk *Use Case* Simpan File

g. *Activity Diagram* Cetak Hasil

Activity Diagram Cetak Hasil menggambarkan proses mencetak hasil dari keluaran *tree* yang terbentuk. Pengguna memilih menu File-Print, selanjutnya sistem akan menampilkan form print dialog dan melanjutkan dengan proses mencetak keluaran.



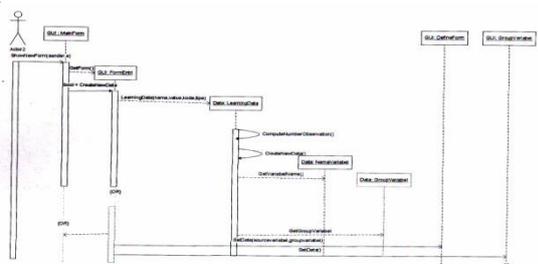
Gambar 3.11. Rancangan *activity diagram* untuk *Use Case* Cetak Hasil

3. Sequence Diagram

Sequence Diagram menggambarkan alur logika prosedural aplikasi. Penggambaran *sequence diagram* didasarkan pada diagram *use case*. Berikut ini merupakan *sequence diagram* untuk masing-masing *use case*.

Sequence Diagram Entri Data

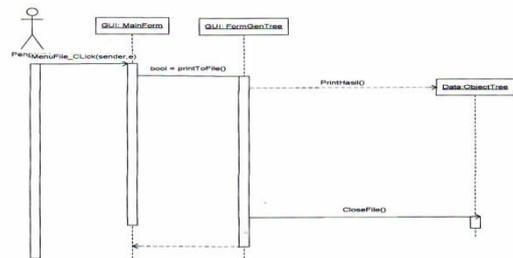
Proses pembuatan data baru dimulai dengan pengguna memilih menu File New pada MainForm yang mengakibatkan MainForm membuat instansiasi objek baru dari kelas Form Entri dan memanggil method *GetForm()*. Proses selengkapnya bisa dilihat pada Gambar 3.12.



Gambar 3.12. *Sequence Diagram* Buat Data

Sequence Diagram Cetak Hasil

Sequence diagram ini menjelaskan proses menyimpan file yang diawali dengan pengguna memilih menu File-Print pada MainForm yang mengakibatkan pemanggilan method *PrintFile ()* pada instance dari kelas *FormGenerateTree*. Diagram selengkapnya bisa dilihat pada Gambar 3.13.



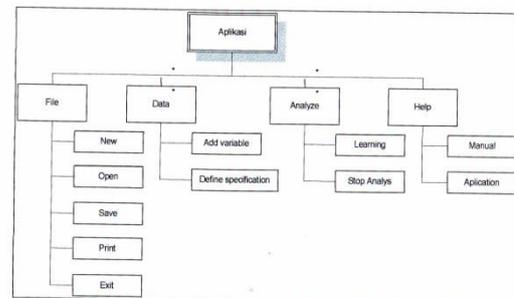
Gambar 3.13. *Sequence Diagram* Cetak Hasil

D. Rancangan Antarmuka

Rancangan antarmuka pengguna (*user interface*) dibuat untuk memberikan kemudahan kepada pengguna dalam berinteraksi dan berkomunikasi dengan sistem. Sistem menerima masukan dan menampilkan keluaran melalui antarmuka yang tampil pada layar monitor. Rancangan antarmuka dalam penelitian ini dibagi menjadi tiga, yaitu rancangan menu, rancangan antarmuka masukan, dan rancangan antarmuka keluaran.

1. Rancangan Menu

Rancangan menu dibuat untuk memudahkan pengguna dalam menggunakan fungsi-fungsi yang ada. Pada tampilan utama terdapat menu-menu pull-down yang dibagi lagi menjadi sub-submenu. Rancangan menu dapat dilihat pada Gambar 3.14

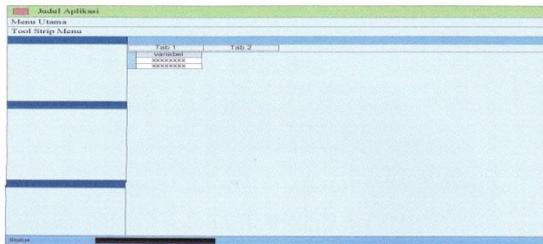


Gambar 3.19. Rancangan Menu Utama

2. Rancangan Antarmuka Masukan

Rancangan masukan dirancang untuk memudahkan pengguna dalam melakukan input data dan load data.

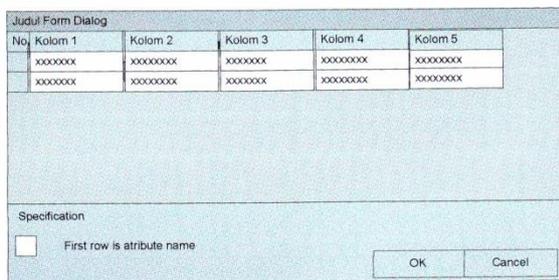
a. Rancangan antarmuka Halaman Utama (Main Form)



Gambar 3.20. Rancangan Antarmuka Halaman Utama

b. Rancangan antarmuka untuk Import Data

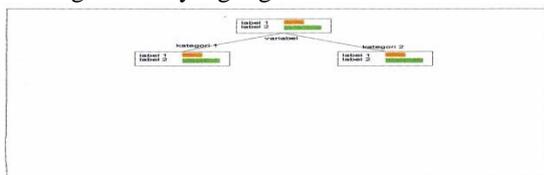
Pengguna dapat menggunakan data yang tersimpan dalam bentuk file eksternal berformat .mdb maupun text file. Antarmuka ini dirancang untuk memfasilitasi pengguna pada saat melakukan import data.



Gambar 3.21. Rancangan Antarmuka untuk Import Data file eksternal .mdb

3. Rancangan antarmuka keluaran

Antarmuka keluaran merupakan container untuk menampilkan tree hasil induksi berdasarkan learning data set yang digunakan.



Gambar 3.26. Rancangan antarmuka Keluaran

IV. SIMPULAN DAN SARAN

A. Simpulan

1. Kematian *neonatum* dipengaruhi oleh faktor sosial biologi ibu hamil, yang terdiri dari variabel tempat tinggal, tingkat pendidikan, tingkat pendapatan keluarga, dan urutan kelahiran. Variabel yang paling mempengaruhi adalah variabel tingkat pendidikan.
2. Aplikasi yang dibangun menggunakan algoritma C4.5 dapat mendukung teknik analisis data mining, khususnya untuk teknik klasifikasi.
3. *Decision Tree* yang dihasilkan masih bersifat statis, yaitu terbatas hanya pada empat variabel

dan maksimal tiga cabang untuk masing-masing node-nya

4. Program aplikasi dirancang dengan berorientasi objek, menggunakan alat pemodelan *Unified Modeling Language* dan diimplementasikan dengan alat bantu bahasa pemrograman *Microsoft Visual Basic .NET* dan editor *Microsoft Visual Studio 2005*

B. Saran

1. Perlu ditambahkan fungsi untuk melakukan pendeteksian validitas set data yang digunakan, misalnya pendeteksian data *outlier*, *missing value*, korelasi antarvariabel, maupun sebaran datanya.
2. Pengembangan sistem selanjutnya diharapkan mampu menganalisis lebih dari empat variabel. Hal ini memungkinkan tindakan analisis memperhatikan faktor lain penyebab kematian *neonatum*.
3. Percabangan yang dihasilkan oleh pohon keputusan (*decision tree*) hanya maksimal tiga *child*, sehingga untuk pengembangan sistem selanjutnya diharapkan aplikasi dapat menangani variabel atribut dengan lebih dari tiga kategori.
4. Aplikasi dapat dikembangkan berbasis web

DAFTAR PUSTAKA

- [1] Ariadni, Ratih dan Isye Arieshanti. 2011. *Implementasi Metode Pohon Keputusan untuk Klasifikasi Data dengan Nilai Fimr Yang Tidak Pasii*. Surabaya, Institut Teknologi Sepuluh November.
- [2] Djaja, Sarimawar. Afifah, Tin. Dan Ahmad Sukroni. 2004. *Peran Faktor Sosio Ekonomi dan Biologi terhadap Kematian Neonatal di Indonesia*. Badan Penelitian dan Pengembangan Kesehatan, Departemen Kesehatan Republik Indonesia.
- [3] Larose, Daniel T. 2005. *Discovering Knowledge in Data. An Introduction to Data Mining*. New Jersey, A John Wiley & Sons, Inc.
- [4] Munawar, 2005. *Pemodelan Visual dengan UML*. Yogyakarta, Penerbit Graha Ilmu.
- [5] Hamilton, Kim. And Russel Miles., 2006. *Learning UML 2.0*. California, O'Reily Media
- [6] <http://en.wikipedia.org/wiki/C4.5algorithm>
- [7] <http://eric.univ-lyon2.fr/~ricco/sipina.html>
- [8] <http://hciteseerx.ist.psu.edu/~decision.htm>
- [9] Tjortjis, Christos and John Keane. 2002. *A Classification Algorithm for Data Mining*.



Manchester, Department of Computation,
UMIST.

- [10] Febri Wulandari, Verdi Yasin, Abdi Moissa (2018) “Perancangan jaringan voip dengan ivr (interactive voice respon) untuk sistem informasi akademik stmik jayakarta menggunakan asterisk” *Journal of Information System, Informatics and Computing*. e-ISS: 2597-3673 (online), p-ISSN: 2579-5201 (print). Vol. 2 No.1 (2018) pp.51-56
- [11] Tri Wahyudi, Zulhalim (2018) “Perancangan aplikasi monitoring angka kredit pejabat fungsional pengendali ekosistem hutan pada ditjen planologi kehutanan dan tata lingkungan kementrian lingkungan hidup dan kehutanan” *Journal of Information System, Applied, Management, Accounting and Research*. e-ISSN: 2598-8719 (online), p-ISSN: 2598-8700 (print), Vol. 2 No. 2 (2018) pp.61-67.