

PENERAPAN ALGORITMA C4.5 PADA SISTEM INFORMASI KLINIK MATA FITUR REGISTRASI PASIEN

¹⁾Anang Andrianto,ST., MT²⁾Zakiyah Qurrotul Aini, ³⁾M. Arief Hidayat,S.Kom., M.Kom.

¹⁾²⁾³⁾Program Studi Sistem Informasi, Universitas Jember,

email: anang.upti@unej.ac.id

Abstract: *Eye Clinic located at bangkalan is a clinic tasked with providing eye health services to the public. One form of service at the eye clinic is the examination of the patient. To perform the examination, patients are required to wait in line according to the queue acquired during the registration. At the eye clinic needed a patient registration system that can predict the patient examination time so that the patient was not long time to wait the queue in place. This research is to design and build a web-based system of patients registration using C4.5 algorithm to predict the patient examination time based on patient complaints. C4.5 algorithm is one of the decision tree algorithm that can be used to predict an event. The generated software has been tested can predict the patient examination time so that patients can wait the queue at home. Rule is generated by an algorithm C4.5 accuracy value of 65.8% and 34.2% error rate so that the algorithm can be used in patient registration feature.*

Keywords: *C4.5algorithm, the decision tree, Eye Clinic, Registration Information System*

1. Pendahuluan

Klinik Mata yang bertempat di Bangkalan, Madura merupakan sebuah klinik yang bekerja dalam bidang kesehatan mata yang bertugas memberikan pelayanan kesehatan mata kepada masyarakat. Klinik mata tersebut melayani pasien dalam bentuk pemeriksaan pasien, tindakan operasi, pemesanan kacamata, dan lain-lain. Selama ini segala kegiatan klinik maupun pengelolaan data pasien dan obat dilakukan secara konvensional. Pasien harus melakukan beberapa proses yang cukup memakan waktu jika akan melakukan tindakan pemeriksaan dan operasi di klinik tersebut.

Proses yang dimaksud diantaranya adalah proses registrasi yang masih dicatat secara konvensional menggunakan prinsip First in First out (FIFO), di mana pasien harus datang sendiri ke klinik kemudian mendaftar dan menunggu antrian untuk diperiksa. Hal tersebut tidak akan menimbulkan masalah jika pasien yang akan berobat pada saat itu tidak terlalu banyak, namun jika pasien pada saat itu cukup banyak maka pasti akan terdapat masalah antrian yang panjang. Hal ini menjadikan ketidaknyamanan pasien.

Pasien dengan keluhan penyakit serius harus mendapatkan penanganan khusus yang pastinya memakan waktu pemeriksaan yang cukup lama sehingga akan berpengaruh kepada waktu tunggu pasien lain yang melakukan antrian. Sedangkan untuk tindakan operasi, selain melalui proses registrasi pasien harus melakukan proses penjadwalan operasi. Penjadwalan operasi harus menyesuaikan waktu pasien dan dokter yang bersangkutan.

Selain registrasi dan penjadwalan pasien, data pasien, rekam medis, data obat, data kacamata, serta data-data transaksi juga masih dikelola secara konvensional. Semakin banyak pasien dan permintaan pelayanan, maka pengelolaan data secara konvensional sudah tidak efisien dan beresiko karena data yang dikelola secara konvensional menggunakan kertas, dan jika salah satu kertas yang berisi data penting tersebut hilang maka akan mempengaruhi data-data lainnya.

Berdasarkan uraian di atas permasalahan yang paling utama adalah pada antrian pasien, oleh karena itu diperlukan suatu pengelolaan data dalam bentuk sistem terkomputerisasi yaitu sistem informasi klinik dokter mata [1]. Pengelolaan data antrian dapat dilakukan

didasarkan pada data pasien yang registrasi dan keluhan yang disampaikan. Prediksi ketepatan waktu kedatangan dapat menggunakan suatu algoritma. Algoritma c4.5 merupakan algoritma yang dapat digunakan untuk memprediksi suatu kejadian, algoritma ini merupakan salah satu algoritma induksi pohon keputusan yang dikembangkan oleh J. Ross Quinlan [2].

Pemanfaatan Algoritma C4.5 pada sistem digunakan pada fitur registrasi pasien untuk proses penentuan atau prediksi waktu penanganan pasien, metode ini menentukan waktu penanganan dengan melihat keluhan yang dialami oleh pasien. Data keluhan tersebut nantinya akan diproses menggunakan decision tree sehingga dapat menghasilkan suatu keputusan waktu penanganan pasien yang diterjemahkan ke dalam bahasa linguistik yaitu, cepat, sedang, dan lama.

Pasien dapat melakukan pendaftaran secara online dengan mengisi data pendaftaran dan keluhan yang dialami, kemudian pasien dapat melihat nomer antrian dan waktu pemeriksaan pada website, sehingga pasien tersebut dapat menunggu di rumah sampai dengan waktu periksa yang telah ditentukan.

Target dari penelitian ini adalah membangun sistem informasi klinik mata dengan fitur registrasi pasien dengan prediksi waktu pemeriksaan pasien menggunakan algoritma c4.5

2. Metode Penelitian

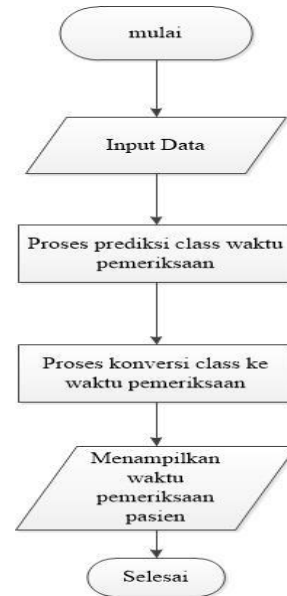
Pendekatan penelitian yang dipilih peneliti untuk menjawab perumusan masalah penelitian yang telah dijelaskan adalah pendekatan penelitian kualitatif. Peneliti menggunakan pendekatan penelitian kualitatif karena hasil dari penelitian ini tidak didapat dari prosedur statistik melainkan dari data-data yang dikumpulkan melalui beberapa cara yaitu wawancara dan survei. Penelitian ini dilakukan di klinik mata dr. Jamaluddin Sp.M. yang beralamat di jl. Bhayangkara M. Hosen 29 Bangkalan, Madura.

Guna memenuhi target peneliti menggunakan beberapa metode yaitu, metode pengumpulan data, metode yang diimplementasikan dalam membangun sistem yaitu algoritma c4.5 guna menentukan waktu pemeriksaan pasien, dan metode *System Development Lyfe Cycle* (SDLC)

model incremental guna membangun sistem secara keseluruhan[3].

Pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan dalam penelitian. Data-data yang dibutuhkan pada penelitian ini seperti data pasien, data rekam medis pasien untuk menentukan atribut dalam pembuatan sistem, alur kerja pada klinik mata untuk menentukan fitur- fitur yang akan dibangun pada sistem, serta data set yang akan digunakan sebagai data training dan data testing dalam penerapan algoritma c4.5 untuk menentukan waktu pemeriksaan pasien.

Guna menentukan prediksi waktu pemeriksaan tersebut perlunya terlebih dahulu dibuat rule yang dihasilkan dari proses datamining dengan menggunakan salah satu *algoritma decision tree* yaitu algoritma c4.5. Flowchart penerapan algoritma c4.5 pada fitur registrasi pasien terdapat pada gambar 1.



Gambar 1 Flowchart penerapan algoritma c4.5 pada fitur registrasi

Terdapat beberapa tahapan, yang pertama adalah input data yang berupa data keluhan yang dialami pasien, usia, serta jenis kelamin pasien. Kemudian data tersebut digunakan dalam proses prediksi class waktu periksa. Ada tiga kategori class waktu periksa yaitu cepat, sedang, dan lama. Setelah ditemukan kategori class yang sesuai, class akan dikonversi kedalam waktu pemeriksaan dalam bentuk menit(cepat = 10

menit, sedang = 20, dan lama = 30), lalu ditentukan waktu pemeriksaan dalam bentuk jam.

Proses prediksi class didasarkan pada rule yang telah dikonversi sebelumnya dari hasil membangun decision tree. Proses pembangunan decision tree pada sistem tersebut meliputi beberapa proses utama yaitu: proses request data, proses pembangunan model (*proses learning*), dan pengklasifikasian data (*proses testing*).

Proses yang pertama dilakukan yaitu proses request data, dimana sistem mendapatkan data untuk diolah, yaitu data training dan data testing beserta informasinya. Informasi yang didapat dari data training yang didapatkan yaitu berupa jumlah record tabel data training, sedangkan informasi yang didapat dari data testing yaitu jumlah record tabel data testing.

Proses selanjutnya yaitu proses learning. Adapun langkah-langkah pada proses learning, yaitu: Sistem mendapatkan data training; Sistem membangun decision tree dari data training dengan algoritma c4.5[4]; Sistem menghasilkan decision tree; Sistem mengkonversi decision tree ke dalam bentuk rule dan menyimpan rule tersebut.

Setelah didapatkan *decision tree*, *decision tree* di konversi ke dalam bentuk *rule*. Jika semua data didalamnya sudah memiliki kelas yang sama maka node dinyatakan sudah murni maka proses pembangunan tree selesai. Jika node belum murni, maka melanjutkan ke proses membagi node sampai dihasilkan kondisi murni.

Proses membuat node diawali dengan mendapatkan data atribut xi, lalu hitung distribusi kelas atribut xi pada node. setelah itu, cek apakah atribut merupakan kategori, jika ya maka hitung nilai entropi menggunakan persamaan[4]

$$H_S(X) = \sum_{i=1}^k P_i H_S(X_i),$$

jika tidak maka tentukan nilai batas (*Threshold*) Z kemudian hitung entropi tiap nilai batas Z. Pilih nilai Z yang memiliki nilai H_{z_i} paling tinggi untuk menjadi nilai batas atribut yang akan dibandingkan dengan atribut lainnya. Setelah mendapat nilai entropi tiap atribut, proses

selanjutnya yaitu menghitung $gain(X_i)$ menggunakan persamaan[4]

$$gain(S) = H(X) - H_S(T)$$

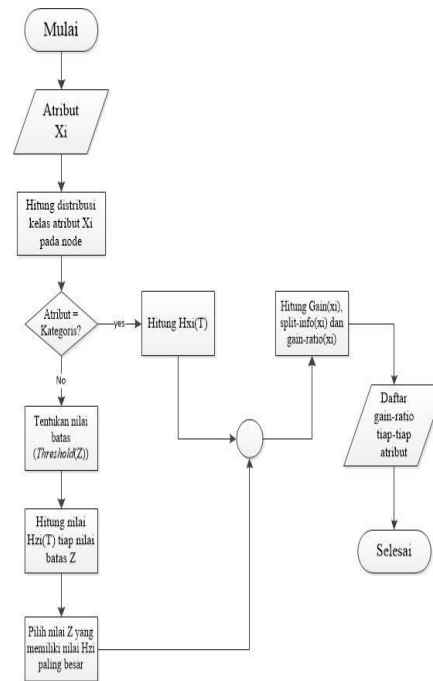
$split-info(X_i)$ menggunakan persamaan [4]

$$Split - info(X)$$

$$= - \sum_{i=1}^n ((|T_i|/|T|) \log_2(|T_i|/|T|))$$

serta $gain-ratio(X_i)$ menggunakan persamaan [4]

$$Gain - ratio(X) = gain(X)/split - info(X).$$



Gambar 3 Flowchart proses membuat node

Contoh proses pembangunan tree dengan menggunakan tabel data training yang dapat dilihat pada Tabel 1.

Contoh perhitungan iterasi 1:

Jumlah record keseluruhan = 15

- Distribusi kelas:

Kelas cepat = 6

Kelas sedang = 6

Kelas lama = 3

Jika probabilitas kelas ditunjukkan dalam tabel 2.

Tabel 1. Contoh data training

No	Usia	J.Kelamin	Mata Merah	Pandangan Kabur	Mata Berair	Silau	Gatal	Nyeri	Kelopak Mata Bengkak	Pandangan Berkabut	Keluar Kotoran (b.a.k.a.)	Tidak dapat membedakan	Pandangan Sempit	Kabur lihat jauh	Kabur lihat dekat	Waktu
1	50	Laki-laki	Ya	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Ya	Cepat
2	49	Perempuan	Ya	Ya	Ya	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Ya	Sedang
3	47	Perempuan	Tidak	Tidak	Ya	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
4	60	Laki-laki	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Cepat
5	13	Perempuan	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Sedang
6	24	Laki-laki	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
7	60	Perempuan	Tidak	ya	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Sedang
8	13	Perempuan	Ya	Tidak	Tidak	Tidak	Tidak	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
9	14	Laki-laki	Ya	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Cepat
10	16	Perempuan	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Sedang
26	85	Laki-laki	Tidak	ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Lama
124	15	Perempuan	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Lama
131	30	Perempuan	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Lama
142	50	Laki-laki	Ya	Tidak	Ya	Tidak	Tidak	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Sedang
145	18	Perempuan	Tidak	Tidak	Ya	Ya	Tidak	Ya	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Sedang

Maka probabilitas kelas dapat ditunjukkan dalam tabel 2.

Tabel 2. Probabilitas Kelas

Kelas	Jumlah	Probabilitaskelas
Cepat	6	6/15
Sedang	6	6/15
Lama	3	3/15

Hasil perhitungan entropi kelas:

$$H(X) = -6/15 \log_2 6/15 - 6/15 \log_2 6/15 - 3/15 \log_2 3/15 = 1,521928095$$

- Menghitung informasi tiap atribut:

- Ditribusi kelas atribut mata merah (X₁) ditunjukkan dalam tabel 3.

Tabel Error! No text of specified style in document..Tabel distribusi kelas atribut mata merah

Outcomes	Cepat	Sedang	Lama	Jumlah records
Ya	1	2	2	5
Tidak	5	4	1	10
Jumlah	6	6	3	15

Perhitungan entropi dan information-gain untuk atribut X₁, sebagai berikut:

$$H_{x1}(X) = 6/15(-3/6 \log_2 3/6) - 2/6 \log_2 2/6 - 1/6 \log_2 1/6 + 9/15 (-3/9 \log_2 3/9) - 4/9 \log_2 4/9 - 2/9 \log_2 2/9 = 1,501955001$$

$$\text{Gain}(X_1) = 1,521928095 - 1,501955001 = 0,019973094$$

Sedangkan perhitungan split-info dan gain-ratio untut atribut x1 sebagai berikut:

$$\text{Split-info}(x1) = -6/15 * (\log_2 6/15) - 9/15 * (\log_2 9/15) = 0,970950594$$

$$\text{Gain-ratio}(X_1) = 0,970950594 / 0,019973094 = 0,020570659$$

Dengan meakukan perhitungan seperti pada perhitungan atribut mata merah pada setiap atribut yang memiliki tipe kategori. Sedangkan atribut yang memiliki tipe numeris, memiliki langkah yang berbeda. Langkah pertama yang dilakukan yaitu mencari nilai batas, dan mencari informasi gain dari tiap-tiap nilai batas.

Contoh perhitungan atribut numeris (umur).

$$\text{Unique value} = 13, 14, 15, 16, 18, 24, 30, 47, 49, 50, 60, 85$$

$$\text{Nilai batas} = 13, 14, 15, 16, 18, 24, 30, 47, 49, 50, 60$$

- Distribusi kelas atribut umur

Distribusi kelas atribut umur ditunjukkan dalam tabel 4.

Tabel 4. Tabel distribusi kelas atribut umur

Nilai batas	Cepat	Sedang	Lama
<=13	1	1	0
>13	5	5	3
<=14	2	1	0
>14	4	5	3
<=15	2	1	1
>15	4	5	2
<=16	2	2	1
>16	4	4	2
<=18	2	3	1
>18	4	3	2
<=24	3	3	1
>24	3	3	2
<=30	3	3	2
>30	3	3	1
<=47	4	3	2
>47	2	3	1
<=49	4	4	2
>49	2	2	1
<=50	5	5	2
>50	1	1	1
<=60	6	6	2
>60	0	0	1

Contoh perhitungan information-gain nilai batas 13 :

$$H_{y \leq 13 \text{ or } y > 13} (X) = 2/15(-1/2 \log_2 1/2 - 1/2 \log_2 1/2) + 13/15 (-5/13 \log_2 5/13 - 5/13 \log_2 5/13 - 3/13 \log_2 3/13) = 1,475436526$$

$$\text{Gain}_{y \leq 13 \text{ or } y > 13} = 1,521928095 - 1,475436526 = 0,046491569$$

Untuk perhitungan gain nilai batas lainnya dihitung sama seperti perhitungan gain nilai

batas 13. Tabel 5 menyajikan hasil perhitungan information-gain tiap nilai batas.

Tabel 5. Tabel hasil perhitungan information-gain tiap nilai batas

Batas	Hx(X)	Gain
y≤13 or y>13	1,475436526	0,046491569
y≤14 or y>14	1,427327302	0,094600793
y≤15 or y>15	1,496273822	0,025654273
y≤16 or y>16	1,521928095	0
y≤18 or y>18	1,501955001	0,019973094
y≤24 or y>24	1,50879563	0,013132465
y≤30 or y>30	1,50879563	0,013132465
y≤47 or y>47	1,618947501	- 0,09701 9406
y≤49 or y>49	1,521928095	0
y≤50 or y>50	1,503677104	0,018250991
y≤60 or y>60	1,352227927	0,169700168

Dari perhitungan diatas, didapat nilai batas dengan nilai information-gain tertinggi 0,169700168 yaitu nilai batas 13. Maka, information-gain dari nilai batas 13 digunakan untuk dibandingkan dengan atribut lain untuk mencari node. Hasil perhitungan information-gain dan gain-ratio dapat dilihat pada tabel 6.

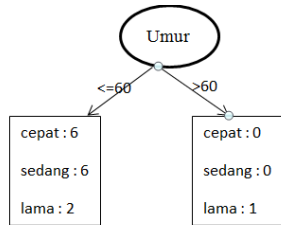
Setelah didapatkan nilai *gain-ratio* tiap-tiap atribut, pilih atribut dengan nilai paling besar.

Tabel 6 Tabel hasil perhitungan information-gain dan gain-ratio iterasi 1

No	Class	Hs(x)	Gain(x)	Split-info(x)	Gain-ratio(x)
1	Mata Merah	1,501955001	0,019973094	0,970950594	0,020570659
2	Pandangan Kabur	1,41461873	0,107309365	0,918295834	0,116857075
3	Mata Berair	1,370950594	0,1509775	0,918295834	0,164410525
4	Silau	1,427327302	0,094600793	0,721928095	0,131039079
5	Gatal	1,428577729	0,093350366	0,353359335	0,264179709
6	Nyeri	1,398118963	0,123809132	0,996791632	0,124207635
7	Kelopak Mata Bengkak	1,475436526	0,046491569	0,566509507	0,082066706
8	Pandangan Berkabut	1,428577729	0,093350366	0,353359335	0,264179709
9	Keluar Kotoran	1,503677104	0,018250991	0,721928095	0,025280898
10	tdk dpt mmbeda warna	1,217542476	0,304385619	0	0
11	pandangan sempit	1,217542476	0,284412525	0	0
12	kabur lihat jauh	1,41992402	0,102004075	0,836640742	0,121920999
13	kabur lihat dekat	1,475436526	0,046491569	0,566509507	0,082066706
14	Umur	1,352227927	0,169700168	0,353359335	0,480248153
15	Jenis Kelamin	1,36196397	0,159964125	0,970950594	0,164750015

Dari perhitungan diatas, didapat *gain-ratio* terbesar pada atribut umur yaitu 0,480248153. Maka atribut umur yang di jadikan node.

Kemudian bagi node berdasarkan atribut yang terpilih. Decision tree yang terbentuk pada iterasi 1 ditunjukkan pada gambar 4.



Gambar 4 Decision tree iterasi 1

Langkah selanjutnya adalah mengecek apakah node sudah murni atau belum. Jika node sudah murni, maka proses berhenti dan decision tree terbentuk. Jika belum, proses membagi node dilakukan kembali secara berulang hingga semua node yang terbentuk menjadi murni dan sudah tidak ada lagi node yang dapat dibagi.

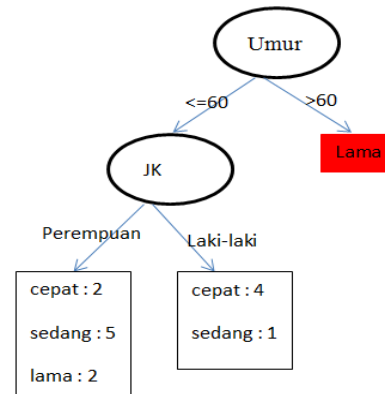
Untuk iterasi dua, hasil pengecekan node > 60 menunjukkan node sudah murni (node memiliki satu kelas yaitu, lama). Untuk node <= 60, node belum murni, karena didalamnya masih terdapat beberapa kelas yang berbeda. Maka, node <= 60 di-split. Untuk mencari atribut yang menjadi node, dilakukan kembali menghitung informasi tiap-tiap atribut yang belum menjadi node. Dengan cara yang sama seperti pada iterasi 1 didapat hasil perhitungan *informastion-gain* dan *gain ratio* tiap atribut pada iterasi 2.

Setelah didapatkan nilai *gain-ratio* tiap-tiap atribut, pilih atribut dengan nilai paling besar. Dari perhitungan diatas, didapat *gain-ratio* terbesar pada atribut jenis kelamin yaitu 0,285178738. Maka atribut jenis kelamin yang di jadikan node.

Kemudian bagi node berdasarkan atribut yang terpilih. Decision tree yang terbentuk pada iterasi 2 ditunjukkan pada gambar 5.

Seperti pada iterasi 1, node dicek apakah node sudah murni atau belum. Jika node sudah murni maka decision tree terbentuk. Jika node belum murni maka proses diulangi sampai node murni dan tidak ada lagi node yang dapat dibagi.

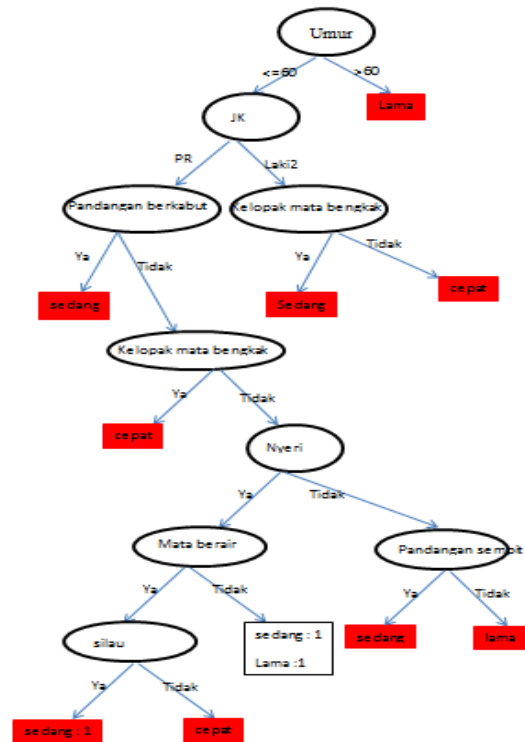
Sehingga decision tree yang terbentuk dari beberapa iterasi setelahnya dapat dilihat pada gambar 6.



Gambar 5. Decision tree iterasi 2

Setelah model tree dibangun, kemudian dikonversi ke dalam bentuk rule. Pembentukan rule ini menggunakan aturan IF-THEN.

Rule yang didapat digunakan untuk memprediksi jika ada data baru yang belum diketahui kelasnya. Dalam sistem informasi klinik mata, rule digunakan pada fitur registrasi pasien untuk menebak atau memprediksi lama penanganan pasien yang kemudian akan di convert menjadi waktu pemeriksaan pasien.



Gambar 6. Decision tree(hasil akhir)

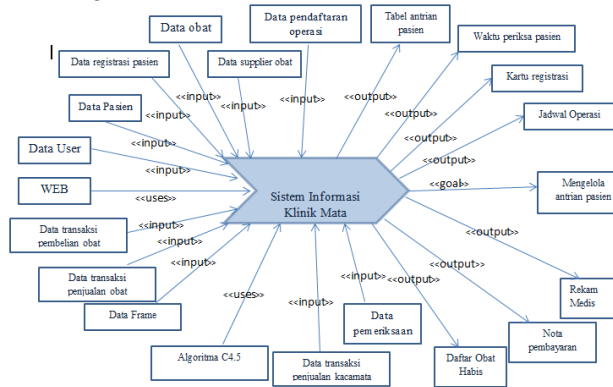
Setelah pembentukan Algoritma c4.5 untuk penentuan waktu pemeriksaan telah didapat maka dilanjutkan dengan tahap pengembangan sistem.

Pengembangan sistem informasi klinik mata ini menggunakan model incremental yang akan di bagi menjadi empat tahap increment dimana di setiap incrementnya dibagi beberapa fitur yang akan diselesaikan. Pada setiap increment terdapat beberapa tahapan, yaitu tahap analisis, desain sistem, coding, dan testing.

Analisis merupakan tahap awal yang dilakukan dalam model incremental. Tahap analisis merupakan tahap pencarian dan analisis kebutuhan.

Desain atau perancangan dalam membangun sistem informasi klinik mata menggunakan Unified Modeling Language (UML) yang dirancang menggunakan konsep Object-Oriented Programming (OOP). Pemodelan UML yang digunakan antara lain :

- Business Process ditunjukkan dalam gambar 7.

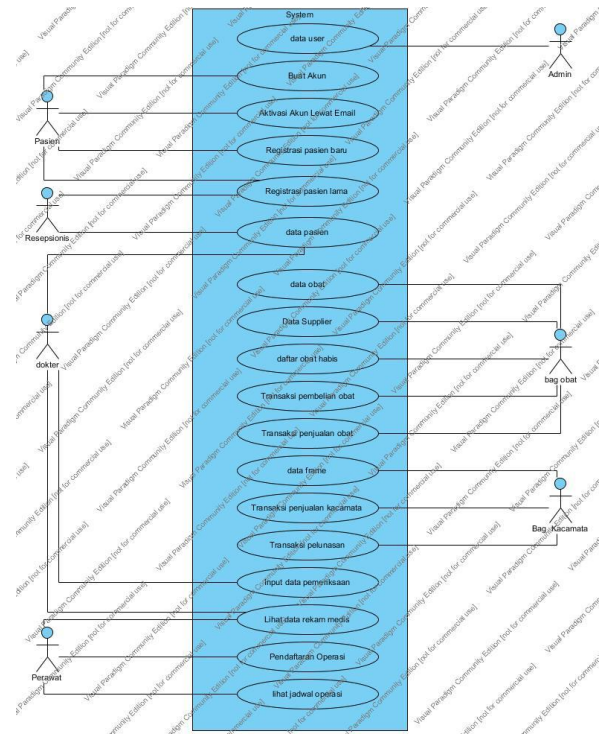


Gambar 7. Business Process Sistem Informasi Klinik Mata

- Use Case Diagram ditunjukkan dalam gambar 8.
- Use Case Scenario
- Sequence Diagram
- Activity Diagram
- Class Diagram ditunjukkan dalam gambar 9.

Pemodelan basis data menggunakan Entity Relationship Diagram (ERD).

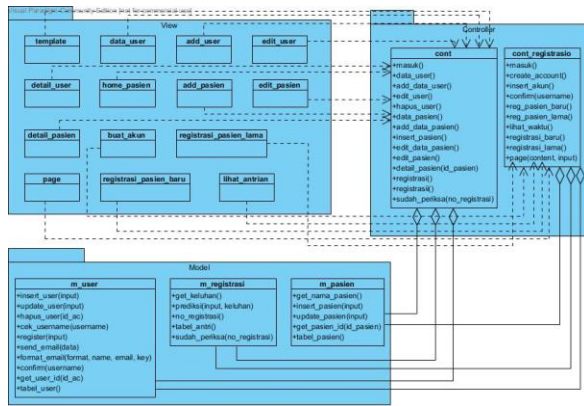
Coding atau penulisan kode program dilakukan menggunakan bahasa pemrograman Hyper Text Pre-Processor (PHP), Hyper Text Markup Language (HTML), Cascading Style Sheet (CSS), dan Javascript. DBMS (Database Management System) yang digunakan adalah MySQL dengan tool XAMPP.



Gambar 8. Usecase Diagram SIKIM

Pengkodean atau coding merupakan proses menulis kode dalam membangun sebuah program. Dalam proses pengkodean sistem informasi klinik mata, penulis menggunakan bahasa pemrograman Page Hyper Text Pre-processor (PHP), Hyper Text Markup Language (HTML), Cascading Style Sheet (CSS), dan Javascript. MySQL sebagai sistem manajemen database. Pengkodean sistem menggunakan konsep object oriented programming (OOP) menggunakan framework codeigniter. Berikut potongan kode program untuk fitur registrasi pasien lama yang terdapat pada kelas controller.

Testing menggunakan *black-box testing* dan *whitebox testing*. *Black-box testing* dilakukan oleh user, user yang melakukan pengujian adalah karyawan dari tempat penelitian. Sedangkan *whitebox testing* dilakukan langsung.



Gambar 9. Class Diagram Tahap Increment 1

3. Hasil dan Pembahasan

Desain yang telah dibuat direalisasikan dengan membuat kode program, berikut beberapa potongan kode program yang paling utama, untuk fitur registrasi pasien lama (controller)

```
function registrasii(){
    $input = array(
        "no_registrasi" => $this->input->post('no_registrasi'),
        "tgl_registrasi" => date("Y-m-d", strtotime($this->input->post('tgl_registrasi'))),
        "id_pasien" => $this->input->post('id_pasien')
    );

    $keluhan = $this->input->post('keluhan');

    $this->m_registrasi->prediksi($input, $keluhan);
    redirect(base_url()
    . "index.php/cont/registrasi?status=sukses");
}
```

Potongan kode program fitur registrasi pasien lama pada kelas controller dengan nama function yang digunakan adalah registrasii(), didalamnya terdapat beberapa variabel dan beberapa kode program untuk memanggil function yang berada pada kelas model m_registrasi yaitu function prediksi(). Potongan kode program pada function prediksi() sebagai berikut:

```
function prediksi($input, $keluhan) {
    $nilai = array();
    $arr_keluhan = array();
    $atr = $this->db->query('select nama_atribut from
    tb_jenisatribut
    . where nama_atribut <> "usia" AND nama_atribut <>
    "jenis_kelamin" ');
    $result = $atr->result_array();
    foreach ($result as $atribut) {
        $sama = false;
        foreach ($keluhan as $k) {
            if ($atribut["nama_atribut"] == $k) {
                $sama = true;
            }
        }
    }
}
```

```
if ($sama) {
    $arr_keluhan[$atribut["nama_atribut"]] = "Ya";
} else {
    $arr_keluhan[$atribut["nama_atribut"]] = "Tidak";
}

//ambil usia
$umurjk = $this->db->query('SELECT id_pasien,
    YEAR(CURDATE()) - YEAR(tgl_lahir) as umur,
    jenis_kelamin
    FROM tb_pasien where id_pasien=' . $input['id_pasien']);
$result1 = $umurjk->row_array();

$arr_keluhan["usia_temp"] = $result1['umur'];
$arr_keluhan["jenis_kelamin"] = $result1['jenis_kelamin'];

$rule = $this->db->get("rule_c45");
$result_rule = $rule->result_array();

$hasil = "";
foreach ($result_rule as $rrule) {
    $id_rule = $rrule["id"];
    $group_rule = $this->db->get_where("rule_penentu_keputusan", array("id_rule" => $id_rule));
    $result_group = $group_rule->result_array();
    $cocok = true;
    foreach ($result_group as $group) {
        foreach ($arr_keluhan as $keluhan => $nilai) {
            if ($group["atribut"] == $keluhan) {
                if ($group["atribut"] == "usia_temp") {
                    $rule_usia = explode(" ", $group["nilai_atribut"]);
                    $standa = $rule_usia[0];
                    $usia = $rule_usia[1];

                    if ($standa == ">") {
                        if ($nilai <= $usia) {
                            $cocok = false;
                            break;
                        }
                    } else if ($standa == "<=") {
                        if ($nilai > $usia) {
                            $cocok = false;
                            break;
                        }
                    } else {
                        if ($group["nilai_atribut"] != $nilai) {
                            $cocok = false;
                            break;
                        }
                    }
                }
            }
        }
    }
    if ($cocok) {
        $hasil = $rrule["keputusan"];
        break;
    }
}

if ($hasil == "") {
    $hasil = "Cepat";
}

//select dari antrian
$selectAntrian = $this->db->query("select selesai as mulai
from "
    . "tb_registrasi where tgl_registrasi = '"
    . $input['tgl_registrasi']
    . "' order by no_registrasi desc limit 1");
$mulai = $selectAntrian->row_array();

if ($mulai['mulai'] == NULL) {
    $waktuMulai = "17:00:00";
} else {
```



```

    $waktuMulai = $mulai['mulai'];
}
//hitung selesainya
$waktuSelesai = explode(":", $waktuMulai);
$TambahMenit = 0;
if ($hasil == "Cepat") {
    $TambahMenit = 10;
} elseif ($hasil == "Sedang") {
    $TambahMenit = 20;
} else { //ini kalo lama
    $TambahMenit = 30;
}
$waktuSelesai[1] += $TambahMenit;
if (($waktuSelesai[1]) >= 60) {
    $waktuSelesai[0] ++;
    $waktuSelesai[1] = 0;
}
$Selesai = implode(":", $waktuSelesai);
$this->db->insert("tb_registrasi", array("no_registrasi" =>
$no_registrasi,
"tgl_registrasi" => $tgl_registrasi, "id_pasien" =>
$id_pasien,
"mulai" => $waktuMulai, "selesai" => $Selesai,
"status" => "Waiting"));
}
    
```

Dari pengkodean di atas ada beberapa logika dalam menentukan waktu pemeriksaan yaitu untuk menebak lama penanganan pasien

kemudian di konversi menjadi waktu pemeriksaan pasien dan penyimpanan data registrasi ke dalam database.

Implementasi Algoritma C4.5 untuk Prediksi Waktu Pemeriksaan Pasien pada fitur registrasi pasien merupakan algoritma c4.5 yang diterapkan pada fitur registrasi pasien untuk menentukan waktu penanganan pasien. Proses request data yang diterapkan pada sistem ditunjukkan pada menu data training dan data testing. Tampilan menu data training dan data testing dapat dilihat pada gambar 10.

Proses learning yaitu mendapat data training. Data training telah dijelaskan pada subbab sebelumnya. Tahap kedua proses ini adalah proses pembangunan tree.

Pada sistem, proses pembangunan tree dilakukan dengan menekan menu proses mining. Kemudian sistem akan melakukan proses pembangunan tree.

NO	NAMA PASIEN	USIA	JENIS KELAMIN	MATA MERAH	PANDANGAN KABUR	MATA BERAIR	SILAU	GATAL	NYERI	KELOPAK MATA BERGAKAR	PANDANGAN BERKABUT	KELUAR KOTORAN BILAK	TIDAK DAPAT MEMBEDAKAN WARNA	PANDANGAN SEMPT	KABUR LIHAT JAUH	KABUR LIHAT DEKAT	WAKTU
1	Sari	50	Laki-laki	Ya	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Ya	Cepat
2	Sunyah	49	Perempuan	Ya	Ya	Ya	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Ya	Sedang
3		47	Perempuan	Tidak	Tidak	Ya	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
4	Haji	60	Laki-laki	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Cepat
5	Windi	13	Perempuan	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Sedang
6	Rohi	24	Laki-laki	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
7	Kubum	60	Perempuan	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Sedang
8	Ainun	13	Perempuan	Ya	Tidak	Tidak	Tidak	Tidak	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat
9	Kuzqiri	14	Laki-laki	Ya	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Cepat
10	Hazamah	16	Perempuan	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Sedang
11	Sahri	53	Laki-laki	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Ya	Sedang
12	Nisuf	31	Laki-laki	Ya	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat

Gambar 10. Tampilan Menu data training dan data testing

Setelah menekan menu proses mining, akan muncul alert, untuk melanjutkan proses mining maka dapat menekan tombol ok. Kemudian sistem akan menjalankan proses mining atau proses pembangunan tree dengan menggunakan algoritma c4.5. langkah-langkah pada proses tersebut didasarkan pada flowchart gambar 4. Langkah pertama adalah menyiapkan data training, kemudian langkah kedua menentukan node atau root. Untuk menentukan node, proses yang pertama adalah menghitung nilai entropi,

menghitung entropi kelas menggunakan persamaan:

$$H(X) = - \sum_i p_i \log_2 p_i.$$

Kemudian menghitung entropi per atribut.

Setelah menghitung entropi, selanjutnya adalah menghitung information gain tiap atribut, Berikut potongan kode program yang menunjukkan perhitungan entropi per atribut, perhitungan information gain, menghitung nilai *split-info* dan *gain-ratio* tiap atribut sebagai berikut:

```
// jika jml kasus = 0 maka entropy = 0
if ($getJumlahKasusTotalEntropy == 0 OR $getJumlahKasusLamaEntropy == 0 OR $getJumlahKasusSedangEntropy == 0 OR $getJumlahKasusCepatEntropy == 0) {
    $getEntropy = 0;
}
// jika jml kasus sama, maka entropy = 1
} else if ($getJumlahKasusLamaEntropy == $getJumlahKasusSedangEntropy AND $getJumlahKasusLamaEntropy == $getJumlahKasusCepatEntropy) {
    $getEntropy = 1;
}
else { // jika jml kasus != 0, maka hitung rumus entropy:
    $perbandingan_lama = $getJumlahKasusLamaEntropy / $getJumlahKasusTotalEntropy;
    $perbandingan_sedang = $getJumlahKasusSedangEntropy / $getJumlahKasusTotalEntropy;
    $perbandingan_cepat = $getJumlahKasusCepatEntropy / $getJumlahKasusTotalEntropy;

    $rumusEntropy = ((-$perbandingan_lama * log($perbandingan_lama,2)) + (-$perbandingan_sedang * log($perbandingan_sedang,2))
    + (-$perbandingan_cepat * log($perbandingan_cepat,2))) / 5;
    $getEntropy = round($rumusEntropy,4); // 4 angka di belakang koma
}
}
```

```
// jumlahkan masing2 inf_gain_temp atribut
$sqlAtributInfGain = mysql_query("SELECT SUM(inf_gain_temp) as inf_gain FROM gain_umur WHERE nilai_atribut = '$sumur'");
while ($rowAtributInfGain = mysql_fetch_array($sqlAtributInfGain)) {
    $getAtributInfGain = $rowAtributInfGain['inf_gain'];
    // echo "atribut = $getAtribut | inf_gain = $getAtributInfGain <br>";

    if ($getAtributInfGain == 0) {
        $getInfGainFix = 0;
    }
    else {
        // hitung inf gain
        $getInfGainFix = round(($getEntropy + $getAtributInfGain),4);
    }
}
```

```
// rumus split info
$getSplitInfo = (($getJumlahKasusTotalEntropy / $getJumlahKasusTotalInfGain) * (log(($getJumlahKasusTotalEntropy / $getJumlahKasusTotalInfGain),2)));
```

```
// rumus gain ratio
$getGainRatio = round(($rowGainRatio['inf_gain'] / $rowGainRatio['split_info']),4);
```

Langkah selanjutnya adalah mencari atribut yang memiliki nilai *gain-ratio* tertinggi. Berikut

potongan kode program yang menunjukkan pencarian *gain-ratio* tertinggi:

```
function getInfGainMax($atribut, $nilai_atribut)
{
    // select inf gain max
    $sqlInfGainMaxAtribut = mysql_query("SELECT distinct atribut FROM mining_c45 WHERE gain_ratio in (SELECT max(gain_ratio) FROM `mining_c45`
    WHERE inf_gain >= 0) LIMIT 1");
    while($rowInfGainMaxAtribut = mysql_fetch_array($sqlInfGainMaxAtribut)) {
        $inf_gain_max_atribut = $rowInfGainMaxAtribut[atribut];
        if (empty($atribut) AND empty($nilai_atribut)) {
            // jika atribut kosong, proses atribut dgn inf gain max pada fungsi loopingMiningC45()
            loopingMiningC45($inf_gain_max_atribut);
        }
        else if (!empty($atribut) AND !empty($nilai_atribut)) {
            // proses atribut dgn inf gain max pada fungsi loopingMiningC45()
            loopingMiningC45($inf_gain_max_atribut);
        }
    }
}
```

Setelah didapatkan atribut dengan *gain-ratio* tertinggi maka atribut tersebut menjadi node. Kemudian node tersebut di split per kelas dan di cek apakah sudah murni atau belum. Jika node sudah murni maka proses berhenti dan membentuk pohon keputusan, sedangkan jika

node belum murni maka sistem akan melakukan looping proses diatas hingga semua node murni atau semua atribut telah digunakan atau tidak ada atribut yang dapat di split. Hasil perhitungan proses mining oleh algoritma c4.5 ditunjukkan pada gambar 11.

> Home		C45 » Perhitungan C4.5									
> Data Training & Testing		Menu: Perhitungan C4.5 Perhitungan Gain Umur									
> C.45 » Perhitungan											
> C.45 » Proses Mining											
> C.45 » Pohon Keputusan											
> Kinerja » Proses Kinerja											
> Kinerja » Perbandingan											
> Logout											

NO	ATRIBUT GAIN RATIO MAX	ATRIBUT	NILAI ATRIBUT	JUMLAH KASUS TOTAL	JUMLAH KASUS LAMA	JUMLAH KASUS SEDANG	JUMLAH KASUS CEPAT	ENTROPY	INFORMATION GAIN	SPLIT INFO	GAIN RATIO
1	pandangan_sempit	Total	Total	164	10	49	105	0.2357			0
2	pandangan_sempit	usia_temp	> 34	47	3	26	18	0.2512	0.0179	0.8643	0.0207
3	pandangan_sempit	usia_temp	<= 34	117	7	23	87	0.2044	0.0179	0.8643	0.0207
4	pandangan_sempit	jenis_kelamin	Laki-laki	67	4	25	38	0.2475	0.0026	0.9757	0.0027
5	pandangan_sempit	jenis_kelamin	Perempuan	97	6	24	67	0.2231	0.0026	0.9757	0.0027
6	pandangan_sempit	mata_merah	Tidak	93	6	28	59	0.2386	0	0.987	0
7	pandangan_sempit	mata_merah	Ya	71	4	21	46	0.2319	0	0.987	0
8	pandangan_sempit	pandangan_kabur	Tidak	129	5	32	92	0.2057	0.0125	0.748	0.0167
9	pandangan_sempit	pandangan_kabur	Ya	35	5	17	13	0.2876	0.0125	0.748	0.0167
10	pandangan_sempit	mata_berair	Tidak	102	3	27	72	0.2024	0.0062	0.9567	0.0065
11	pandangan_sempit	mata_berair	Ya	62	7	22	33	0.274	0.0062	0.9567	0.0065

Gambar 11. Tampilan hasil perhitungan proses mining dengan algoritma c4.5

Hasil dari proses mining atau pembangunan treed adalah pohon keputusan yang dapat dilihat pada gambar 12.

> C.45 » Proses Mining		Pohon Keputusan:									
> C.45 » Pohon Keputusan		<pre> pandangan_sempit = Tidak (Lama = 9, Sedang = 46, Cepat = 105) : ? silau = Tidak (Lama = 9, Sedang = 38, Cepat = 94) : ? pandangan_berkabut = Tidak (Lama = 8, Sedang = 34, Cepat = 93) : ? kabur_lihat_dekat = Tidak (Lama = 8, Sedang = 32, Cepat = 91) : ? kabur_lihat_jauh = Tidak (Lama = 6, Sedang = 28, Cepat = 90) : ? usia_temp = > 48 (Lama = 2, Sedang = 11, Cepat = 4) : Sedang usia_temp = <= 48 (Lama = 4, Sedang = 17, Cepat = 86) : Cepat kabur_lihat_jauh = Ya (Lama = 2, Sedang = 4, Cepat = 1) : ? usia_temp = > 15 (Lama = 2, Sedang = 1, Cepat = 1) : Lama usia_temp = <= 15 (Lama = 0, Sedang = 3, Cepat = 0) : Sedang kabur_lihat_dekat = Ya (Lama = 0, Sedang = 2, Cepat = 2) : ? mata_berair = Tidak (Lama = 0, Sedang = 1, Cepat = 1) : Cepat mata_berair = Ya (Lama = 0, Sedang = 1, Cepat = 1) : Cepat pandangan_berkabut = Ya (Lama = 1, Sedang = 4, Cepat = 1) : ? mata_merah = Tidak (Lama = 1, Sedang = 3, Cepat = 1) : ? jenis_kelamin = Laki-laki (Lama = 0, Sedang = 2, Cepat = 1) : Sedang jenis_kelamin = Perempuan (Lama = 1, Sedang = 1, Cepat = 0) : Sedang mata_merah = Ya (Lama = 0, Sedang = 1, Cepat = 0) : Sedang silau = Ya (Lama = 0, Sedang = 8, Cepat = 11) : ? jenis_kelamin = Laki-laki (Lama = 0, Sedang = 4, Cepat = 5) : Cepat jenis_kelamin = Perempuan (Lama = 0, Sedang = 4, Cepat = 6) : Cepat pandangan_sempit = Ya (Lama = 1, Sedang = 3, Cepat = 0) : ? jenis_kelamin = Laki-laki (Lama = 0, Sedang = 2, Cepat = 0) : Sedang jenis_kelamin = Perempuan (Lama = 1, Sedang = 1, Cepat = 0) : Sedang </pre>									
> Kinerja » Proses Kinerja											
> Kinerja » Perbandingan											
> Logout											

Gambar 12. Tampilan hasil pohon keputusan

Pohon keputusan tersebut kemudian dikonversi menjadi rule yang berdasar pada aturan if-then.

Rule yang dihasilkan sistem dapat dilihat pada gambar 13.

Rule:	
<pre> 1. if (pandangan_sempit == Tidak AND silau == Tidak AND pandangan_berkabut == Tidak AND kabur_lihat_dekat == Tidak AND kabur_lihat_jauh == Tidak AND usia_temp == > 48) then Sedang (id = 6) 2. if (pandangan_sempit == Tidak AND silau == Tidak AND pandangan_berkabut == Tidak AND kabur_lihat_dekat == Tidak AND kabur_lihat_jauh == Tidak AND usia_temp == <= 48) then Cepat (id = 7) 3. if (pandangan_sempit == Tidak AND silau == Tidak AND pandangan_berkabut == Tidak AND kabur_lihat_dekat == Tidak AND kabur_lihat_jauh == Ya AND usia_temp == > 15) then Lama (id = 9) 4. if (pandangan_sempit == Tidak AND silau == Tidak AND pandangan_berkabut == Tidak AND kabur_lihat_dekat == Tidak AND kabur_lihat_jauh == Ya AND usia_temp == <= 15) then Sedang (id = 10) 5. if (pandangan_sempit == Tidak AND silau == Tidak AND pandangan_berkabut == Tidak AND kabur_lihat_dekat == Ya AND mata_berair == Tidak) then Cepat (id = 12) 6. if (pandangan_sempit == Tidak AND silau == Tidak AND pandangan_berkabut == Tidak AND kabur_lihat_dekat == Ya AND mata_berair == Ya) then Cepat (id = 13) 7. if (pandangan_sempit == Tidak AND silau == Tidak AND pandangan_berkabut == Ya AND mata_merah == Tidak AND jenis_kelamin == Laki-laki) then Sedang (id = 16) 8. if (pandangan_sempit == Tidak AND silau == Tidak AND pandangan_berkabut == Ya AND mata_merah == Tidak AND jenis_kelamin == Perempuan) then Sedang (id = 17) 9. if (pandangan_sempit == Tidak AND silau == Tidak AND pandangan_berkabut == Ya AND mata_merah == Ya) then Sedang (id = 18) 10. if (pandangan_sempit == Tidak AND silau == Ya AND jenis_kelamin == Laki-laki) then Cepat (id = 20) 11. if (pandangan_sempit == Tidak AND silau == Ya AND jenis_kelamin == Perempuan) then Cepat (id = 21) 12. if (pandangan_sempit == Ya AND jenis_kelamin == Laki-laki) then Sedang (id = 23) 13. if (pandangan_sempit == Ya AND jenis_kelamin == Perempuan) then Sedang (id = 24) </pre>	

Gambar 13. Tampilan rule yang dihasilkan sistem

Terdapat dua belas rule yang dihasilkan oleh proses pembangunan tree, yang nantinya akan digunakan pada proses prediksi pada fitur registrasi pasien.

Pengklasifikasian (Testing)

Setelah rule terbentuk dari proses learning, selanjutnya dilakukan proses testing. Proses testing yaitu proses pengklasifikasian data testing yang terbentuk. Hasil dari proses testing yaitu

kelas baru untuk tiap data testing yang kemudian akan dibandingkan dengan kelas lama dari data testing, untuk mendapatkan akurasi rule yang terbentuk.

Pada sistem, untuk melakukan proses testing user dapat memilih menu proses kinerja dan untuk melihat hasil testing user memilih menu perbandingan. Tampilan menu perbandingan dapat dilihat pada gambar 14.

NO	NAMA PASIEN	USIA	JENIS KELAMIN	MATA MERAH	PANDANGAN KABUR	MATA BERAIR	SELAU	GATAL	NYERI	KELOMPOK MATA BENGKAK	PANDANGAN BERKABUT	KELUAR KOTORAN BELEK	TIDAK DIPAT HEMBEDAKAN WARNA	PANDANGAN SEMPIT	KABUR LIHAT JAUH	KABUR LIHAT DEKAT	KEPUTUSAN ASLI	KEPUTUSAN C4.5 10 RULE C4.5
1	Sumiyah	49	Perempuan	Ya	Ya	Ya	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Ya	Sedang	Cepat 21
2		47	Perempuan	Tidak	Tidak	Ya	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat	Sedang 6
3	Heji	60	Laki-laki	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Cepat	Sedang 10
4	Windi	13	Perempuan	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Sedang	Sedang 10
5	Rohli	24	Laki-laki	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat	Cepat 20
6	Kulsum	60	Perempuan	Tidak	ya	Tidak	Tidak	Tidak	Tidak	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Sedang	Cepat 7
7	Amun	13	Perempuan	Ya	Tidak	Tidak	Tidak	Tidak	Ya	Ya	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Cepat	Cepat 7

Gambar 14. Tampilan menu perbandingan

Pada tabel tersebut terdapat data testing, keputusan asli dari data testing serta keputusan atau kelas baru yang didapat dari rule yang terbentuk. Proses pengklasifikasian atau testing dilakukan untuk mengukur kinerja atau mengevaluasi dari proses pembentukan tree dengan menggunakan algoritma c4.5. untuk

mengevaluasi hasil testing, penulis menggunakan confusion matrix[6]. Data set yang digunakan dalam pembangunan tree untuk menentukan waktu penanganan pasien memiliki tiga kelas yaitu cepat, sedang, dan lama. Maka confusion matrix hasil klasifikasi waktu penanganan pasien dapat dilihat pada tabel 7.

Tabel 7. Confussion Matrix Hasil Klasifikasi Waktu Penanganan Pasien

Correct Classified / Keputusan Asli / Kelas Asli	Classed As / Keputusan C4.5 / Kelas Baru		
	Cepat	Sedang	Lama
Cepat	100	3	0
Sedang	40	6	2
Lama	7	3	0

Tabel 7 merupakan confusion matrix hasil klasifikasi waktu penanganan pasien. Jumlah record yang memiliki keputusan asli dan keputusan c4.5 benar ada 106 data, sedangkan record yang memiliki keputusan asli dan keputusan c4.5 berbeda ada 55 data. Dengan mengevaluasi decision tree menggunakan confusion matrix akan menghasilkan nilai accuracy dan error rate.

Hasil perhitungan accuracy dan error rate untuk evaluasi hasil klasifikasi waktu penanganan pasien.

$$Accuracy = \frac{100 + 6 + 0}{100 + 6 + 0 + 40 + 7 + 3 + 3 + 0 + 2} \times 100\% = \frac{106}{161} \times 100\% = 65.8\%$$

$$Error\ rate = \frac{3 + 0 + 2 + 40 + 7 + 3}{100 + 6 + 0 + 40 + 7 + 3 + 3 + 0 + 2} \times 100\% = \frac{55}{161} \times 100\% = 34.2\%$$

Berdasarkan hasil perhitungan diatas, nilai accuracy adalah 65.8% yang berarti bahwa jumlah record data yang diklasifikasikan dengan benar adalah 65.8 %. Sedangkan nilai error adalah 34.2%. Kesalahan prediksi tertinggi berada pada keputusan asli ‘sedang’ dan keputusan c4.5 cepat yaitu mencapai angka

empat puluh. Hal tersebut bukan kesalahan yang fatal karena pasien akan mendapatkan waktu pemeriksaan sepuluh menit lebih cepat sehingga tidak akan merugikan proses antrian pasien. Sedangkan apabila terjadi kesalahan sebaliknya atau seperti keputusan asli 'cepat' dan keputusan c4.5 'lama' maka akan merugikan proses antrian pasien karena waktu yang didapatkan pasien menjadi sepuluh menit atau tiga puluh menit lebih lama.

4. Simpulan

Simpulan dari sistem yang dibangun pada penelitian ini adalah sebagai berikut:

- Sistem informasi klinik mata dibangun sesuai kebutuhan dari objek penelitian dan dapat membantu kegiatan pengelolaan data yang terdapat pada klinik mata tersebut.
- Fitur registrasi pasien yang dibangun dengan menerapkan algoritma c4.5 menggunakan lima belas atribut yaitu usia, jenis kelamin, dan tiga belas jenis keluhan yang sering terjadi. Penerapan algoritma c4.5 dapat memprediksi waktu pemeriksaan pasien sehingga pasien tidak menunggu lama di klinik mata tersebut untuk mendapatkan giliran periksa.
- Rule yang dihasilkan oleh algoritma c4.5 memiliki nilai accuracy sebesar 65,8% dan error rate 34,2% sehingga algoritma ini dapat digunakan dalam fitur registrasi pasien.

Adapun saran – saran dari penelitian ini guna meningkatkan hasil yang dicapai, beberapa saran untuk pengembangan penelitian ini diantaranya:

- Melakukan percobaan dengan menggunakan jumlah data training yang lebih banyak sehingga hasil yang dicapai lebih akurat.
- Melakukan penelitian yang sama dengan menggunakan algoritma atau metode yang berbeda atau mengkombinasi algoritma c4.5 dengan algoritma lainnya.
- Menambahkan fitur pengelolaan laporan keuangan pada sistem informasi klinik mata.

Daftar Pustaka

- [1] R. Victor Imbar dan Y. Kurniawan, "Perancangan Sistem Informasi Pelayanan Medis Rawat Jalan Poliklinik Kebidanan dan Kandungan pada RSUD Kota Batam," *Jurnal Sistem Informasi*, pp. Vol.7, No.1, Maret 2012: 53 - 67, 2012.
- [2] A. Andriani, "Sistem Pendukung Keputusan Berbasis Decision Tree Dalam Pemberian Beasiswa Studi Kasus: AMIK "BSI YOGYAKARTA"," *SENTIKA 2013*, p. 165, 2013.
- [3] R. A.S. dan M. Shalahuddin, Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek), Bandung: Modula, 2011.
- [4] D. Larose, *Discovering Knowledge in Databases*, New Jersey: John Wiley and Sons Inc, 2005.
- [5] M. Kantardzic, *Data Mining: Concepts, Models, Methods, and Algorithms*, New York: John Wiley & Sons, INC., 2003.
- [6] F. Gorunescu, *Data Mining: Concepts, Models and Techniques*, Verlag Berlin Heidelberg: Springer, 2011.
- [7] S. Dua dan X. Du, *Data Mining and Machine Learning in Cybersecurity*, Taylor & Francis Group, 2011.
- [8] J. Han dan M. Kamber, *Data Mining Concept and Tehniques*, San Francisco: Morgan Kauffman, 2006.
- [9] D. T. Larose, *Discovering Knowledge In Data: An Introduction to Data Mining*, Canada: John Wiley & Sons, Inc, 2005.
- [10] S. M. Proboyekti, "Software Process Model I," 1 Agustus 2008. [Online]. Available: <http://www.scribd.com/doc/192850316/Software-Process>. [Diakses 18 Oktober 2014].
- [11] C. Mega, "megaboy16.blogspot.com," 12 Maret 2011. [Online]. Available: <http://megaboy16.blogspot.com/2011/03/model-incremental.html>. [Diakses 18 Agustus 2014].