

# Analisis Estimasi Waktu Antrian dengan menggunakan Markov Chain dan Algoritma PageRank

<http://dx.doi.org/10.28932/jutisi.v5i3.1990>

Cato Chandra<sup>#1</sup>, David Sanjaya<sup>#2</sup>, Julio Narabel<sup>#3</sup>, Nucky Vilano<sup>#4</sup>, Setia Budi <sup>#5</sup>

<sup>#</sup>Magister Ilmu Komputer, Universitas Kristen Maranatha  
Jl. Prof. Drg. Surya Sumantri, MPH No.65, Bandung

<sup>1</sup>mi1879002@student.it.maranatha.edu

<sup>2</sup>mi1879012@student.it.maranatha.edu

<sup>3</sup>mi1879003@student.it.maranatha.edu

<sup>4</sup>mi1879006@student.it.maranatha.edu

<sup>5</sup>setia.budi@it.maranatha.edu

**Abstract** — Along with the increasingly rapid development of technology, especially in the field of computers, ways to overcome the problem of patient queues have been developed. One of them is the use of a mobile application to get a time estimate until a patient gets a turn to consult with a doctor. Many industries still use manual methods to overcome this queue problem. Based on this fact, this research has aimed to get time estimates for the patients so that time can be more efficient.

**Keywords** — Markov; Pagerank; application; time; queue; estimation.

## I. PENDAHULUAN

### A. Latar Belakang

Beberapa literatur[8][9] yang membahas mengenai permasalahan antrian di rumah sakit menyebutkan bahwa antrian panjang disebabkan oleh adanya peningkatan kedatangan (*flow rate*) pasien-mendaftar, tidak seimbang dengan tingkat kecepatan pelayanan (*service rate*) dokter[1]. Berdasarkan asumsi yang menyatakan bahwa satuan waktu yang diperlukan oleh pasien hingga mendapatkan pelayanan dokter merupakan waktu kumulatif[2]. Dengan demikian, waktu terlama akan dialami oleh pasien yang datang paling akhir. Hal ini terjadi secara terus-menerus berdasarkan tetapan waktu yang ditetapkan dokter untuk melayani pasien dalam kurun waktu yang sudah ditentukan oleh rumah sakit. Lama waktu kumulatif yang dialami pasien merupakan kerugian waktu serta menurunkan minat pasien untuk melakukan konsultasi media[2].

Selain antrian, perlu diperhitungkan estimasi waktu menunggu yang sebenarnya tidak konstan karena terdapat beberapa faktor acak yang mempengaruhi hal tersebut. Estimasi waktu menunggu dapat lebih diandalkan lagi jika

ketepatan waktu yang diestimasikan sudah berdasarkan pada perhitungan yang melibatkan faktor acak tersebut.

Topik yang diangkat dalam penelitian ini adalah sistem antrian yang dilengkapi dengan analisis perhitungan estimasi waktu menggunakan Markov Chain dan algoritma PageRank sebagai alternatif solusi untuk membangun sistem pendaftaran dan antrian yang praktis dengan tujuan mengurangi kerugian waktu dan biaya yang ditimbulkan pada sistem yang berlaku saat ini.

### B. Rumusan Masalah

Berikut merupakan dua hal yang menjadi fokus permasalahan dalam *paper* ini:

1. Dibutuhkan asumsi PageRank untuk menghitung estimasi waktu dengan bobot yang sama rata.
2. Faktor acak yang mempengaruhi estimasi waktu antrian, antara lain:
  - Antrian pasien kondisi darurat yang memiliki kepentingan lebih tinggi daripada antrian utama.
  - Pembatalan antrian yang disebabkan ketidakhadiran pihak terkait.

### C. Tujuan

Adapun tujuan pembangunan sistem antrian pasien dapat dirumuskan sebagai berikut:

1. Mendapatkan estimasi waktu yang dapat diandalkan, dengan memperhitungkan faktor acak yang mempengaruhi antrian menggunakan asumsi bobot sama rata (*PageRank*).
2. Membuat aplikasi *mobile* untuk sistem antrian praktek dokter dilengkapi dengan perhitungan estimasi waktu.
3. Mengimplementasikan konsep *Markov Chain* dalam perhitungan estimasi waktu antrian.

## II. KAJIAN PUSTAKA

### A. Markov Chain

Markov Model merupakan model yang mendeskripsikan perpindahan antar state berdasarkan probabilitas, yang berarti kondisi masa depan yang akan dituju ditentukan dari probabilitas transisi antar state, bukan dari determinitas. Setiap busur antar state berisi probabilitas yang mengindikasikan kemungkinan jalur tersebut akan diambil. Markov Model atau biasa disebut Markov Chain bermanfaat untuk menghitung probabilitas urutan kejadian yang dapat diamati [3][5].

### B. PageRank

PageRank, memiliki konsep dasar yang sama dengan *link popularity*, tetapi bukan hanya memperhitungkan “jumlah” *inbound* dan *outbound link*. Pendekatan yang digunakan adalah sebuah halaman akan dianggap penting jika halaman lain memiliki *link* ke halaman tersebut. Sebuah halaman juga akan menjadi semakin penting jika halaman lain yang memiliki ranking (PageRank) tinggi mengacu ke halaman tersebut.

Dengan pendekatan yang digunakan PageRank, proses terjadi secara rekursif dimana sebuah ranking akan ditentukan oleh ranking dari halaman web lain yang memiliki link ke halaman tersebut. Proses ini merupakan suatu proses yang berulang (rekursif). Di dunia maya, ada jutaan bahkan milyaran halaman web. Oleh karena itu sebuah ranking halaman web ditentukan oleh struktur *link* dari keseluruhan halaman web yang ada di dunia maya. Sebuah proses yang sangat besar dan kompleks[4][6][12].

Berdasarkan artikel jurnal[10] dikatakan bahwa algoritma PageRank merupakan algoritma yang dapat bekerja dalam konteks umum, yaitu tidak hanya dapat digunakan untuk keperluan menghitung ranking halaman web.

Penggunaan Markov Chain digabungkan dengan algoritma PageRank (yang telah dimodifikasi) pernah dilakukan dalam bidang biologi terkait jaringan biologis untuk mencari pasangan protein yang cocok di antara kedua jaringan tersebut[11]. Artikel jurnal[11] mengatakan bahwa penggabungan algoritma PageRank dan representasi Markov mengungguli algoritma-algoritma lainnya (yang berhubungan dengan bidang jaringan biologis).

### C. REST API

REST (Representational State Transfer), menjelaskan seperangkat prinsip arsitektur dimana data dapat dikirim melalui antarmuka standar (seperti HTTP)[13]. REST tidak mengandung lapisan olah pesan tambahan dan berfokus pada aturan desain untuk membuat layanan tanpa state. Klien dapat mengakses sumber daya menggunakan URI yang unik dan setelahnya, representasi dari sumber daya dikembalikan ke pengguna. Dengan setiap representasi sumber daya baru, klien dikatakan mentransfer status. Saat mengakses sumber daya REST dengan protokol HTTP, URL sumber daya berfungsi sebagai pengidentifikasi

sumber daya dan GET, PUT, DELETE, POST dan HEAD adalah operasi HTTP standar untuk dilakukan pada sumber daya itu.

### D. MongoDB

MongoDB merupakan basis data berbasis dokumen yang terdistribusi dan dibangun untuk pengembang aplikasi modern. MongoDB menyimpan data dalam bentuk dokumen yang mirip JSON.

Kelebihan yang dimiliki MongoDB :

1. Rich JSON Documents, cara paling alami dan produktif untuk bekerja dengan data. Memungkinkan untuk pembuatan skema yang fleksibel dan dinamis.
2. Powerful Query Language, memiliki format yang dinamis, dimana setiap dokumen dapat menyimpan data yang beragam.
3. Mendukung semua kelebihan yang ditawarkan oleh relational database.

Menggunakan NoSQL yang lebih cocok digunakan untuk membangun aplikasi modern, dikarenakan relational database tidak dirancang untuk mengatasi skala dan kemampuan untuk menghadapi aplikasi modern, juga tidak dibangun untuk mengambil keuntungan dari penyimpanan komoditas dan kekuatan pemrosesan yang tersedia saat ini.

### E. NodeJS

NodeJs[7] adalah sebuah platform yang dibuat dengan JavaScript dan runtime Chrome untuk membangun aplikasi network yang cepat dan terukur. NodeJs menjalankan aplikasi berdasarkan peristiwa (event-driven) dan non-blocking I/O yang membuat aplikasi NodeJs lebih cepat, ringan, efisien dan cocok sekali untuk aplikasi dengan transaksi data yang besar. Sama seperti Apache, NodeJs juga merupakan aplikasi yang digunakan pada sisi server (server-side)

### F. Flutter

Flutter merupakan Framework Open Source untuk Mobile Application yang diciptakan oleh Google dengan menggunakan bahasa pemrograman C, C++, Dart, dan Skia (2D Rendering Engine). Flutter digunakan dalam pengembangan aplikasi untuk sistem operasi Android dan iOS, terutama untuk optimisasi pembuatan 2D Mobile Apps yang dapat dijalankan pada Android dan iOS, serta menjadi metode utama untuk membuat aplikasi Google Fuchsia.

Komponen utama Flutter :

1. Flutter Engine, ditulis dengan menggunakan bahasa pemrograman C++. Flutter Engine memberikan dukungan rendering tingkat rendah dengan menggunakan library graph Skia milik Google dan juga dapat berinteraksi dengan perangkat pengembangan perangkat lunak seperti yang disediakan oleh Android dan iOS.
2. Foundation Library, ditulis dengan bahasa pemrograman Dart. Foundation Library

menyediakan fungsi dan class dasar yang digunakan untuk membangun aplikasi menggunakan Flutter, seperti API untuk berkomunikasi dengan engine. Widget, merupakan blok bangunan dasar dari User Interface aplikasi Flutter. Framework Flutter berisi dua set Widget yang disesuaikan dengan rancangan bahasa tertentu. Widget Material Design menerapkan rancangan bahasa Google dan Widget Cupertino meniru rancangan iOS milik Apple.

### III. PERANCANGAN

#### A. Sumber Data

Sumber data diperoleh dengan menggunakan data yang diambil secara acak untuk penamaan yang bersumber dari kumpulan data yang diambil dari internet. Data-data tersebut digabungkan ke dalam sebuah file master di Excel dengan format penamaan setiap lembarannya sebagai berikut: NamaDepan, NamaBelakang, GelarDepan, GelarBelakang, JalanDepan, Jalan Tengah, JalanAkhir. File tersebut digunakan untuk mengolah data secara acak untuk membuat / generate Daftar Nama, dengan mengacak posisi

dari daftar NamaDepan, NamaBelakang. Apabila ingin menggunakan gelar, maka dapat dilakukan dengan cara memproses data-data tersebut dengan menggunakan kombinasi dari daftar GelarDepan, NamaDepan, NamaBelakang, dan GelarBelakang. Dalam pengolahan email, dilakukan pengolahan data ulang dari hasil generate Nama, dengan pemberian titik di tengah dan penambahan variabel "@uas.com" pada bagian akhir. Dalam pengolahan / generate password pengguna, digunakan komponen bcrypt yang disediakan oleh Bahasa pemrograman Python. Dalam pengolahan / generate data tanggal dan jam, digunakanlah komponen regex, yaitu exrex, dengan format tanggal dan waktu.

Sumber data yang diperlukan untuk menunjang aplikasi ini antara lain data dokter, lokasi praktek dokter / rumah sakit, serta data user (pihak pasien). Secara khusus, wilayah akan dibatasi untuk kota Bandung saja. Data dengan jumlah 100 pasien/user dan 100 dokter dapat dilihat pada Tabel I, Tabel II dan Tabel III.

TABEL I  
CONTOH DATA USER YANG DIGUNAKAN DALAM APLIKASI YANG DIKEMBANGKAN

_id	role	email	password	name
NaN	User	Balakosa.Monginsidi@uas.com	b'2b12\$4RmQ0ca.XpapC4ISAqdGK.gz1w6t3rPOynC1....	Balakosa Mongonsidi
NaN	User	Gavin.Terok@uas.com	b'2b12\$V8qHGHtg34hrt2LJAb9Flewin81nMqegM1ZT...	Gavin Terok
NaN	User	Rizqullah.Hutabarat@uas.com	b'2b12\$1MRJ8UVVZYKpWwuo4fgdauWyRy9B2NKqjODil...	Rizqullah Hutabarat
NaN	User	Neila.Mulawarman@uas.com	b'2b12\$M1.LLezfFsFtwlRRwjot3ebR1hVoqpt.4U7N7...	Neila Mulawarman
NaN	User	Dakari.Kindangen@uas.com	b'2b12\$8Sihllqo3uJJv6Atl5AzeyC5ZxeLI7r8jCZ...	Dakari Kindangen

TABEL II  
CONTOH DATA PASIEN YANG DIGUNAKAN DALAM APLIKASI YANG DIKEMBANGKAN

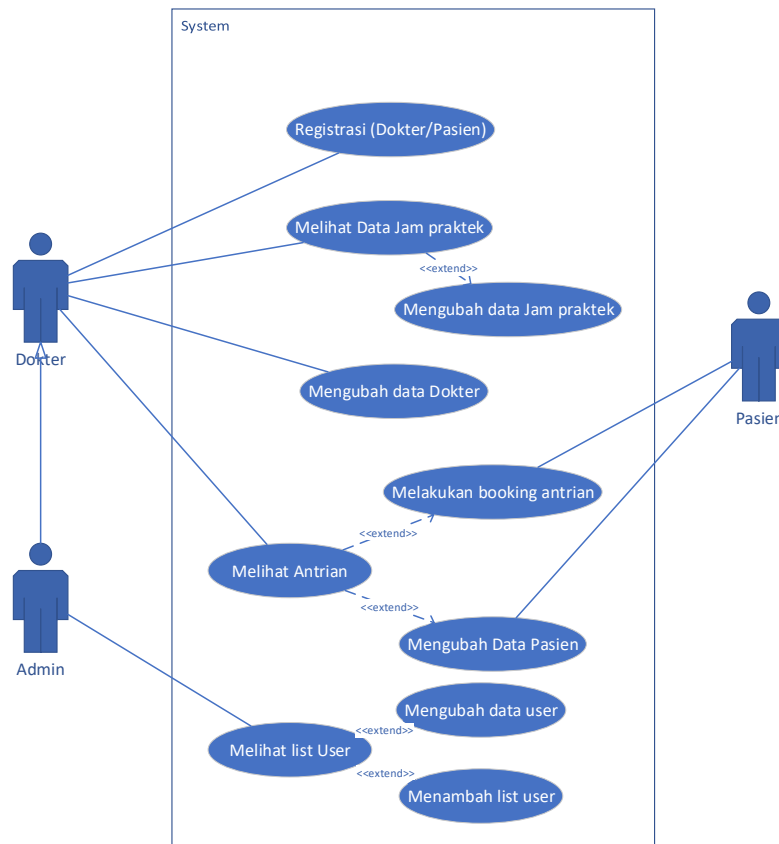
No	Nama	Alamat	Tanggal_Lahir	No_KTP
1	Balakosa Mongonsidi	JL. Sepak Bola Blok 15	25/07/1983	7902250719835461
2	Gavin Terok	Ds. Cidurian Baru No.34	22/04/1998	790220419984932
3	Rizqullah Hutabarat	Ds. Taruna Parahyangan No.80	25/11/1981	7902251119817806
4	Neila Mulawarman	Kompleks Perumahan Citarip kidul Blok 78	23/04/1983	7902230419830512
5	Dakari Kindangen	Komplek Perumahan Pager Gunung Blom 78	23/11/1989	7902231119897069

TABEL III  
CONTOH DATA DOKTER YANG DIGUNAKAN DALAM APLIKASI YANG DIKEMBANGKAN

No	Nama	Alamat	Tanggal_Lahir	Kode_Mitra
1	Prof.Dr.dr. Kawena Turangan Sp.Ok	Jl. Lio Genteng Blok 31	29/09/1996	6163087893692
2	dr. Shuoue Andy Sp.OT	Jln. Pasundan No. 62	09/05/1982	9067801072289
3	Dr.dr. Tansy Liem Sp.Ger	Jl. Batu Kencana Blok 50	18/07/1987	1409123394128
4	dr. Sherwin Hansyi	Jl. Rotgen Blok 12	26/07/1999	9780544551624
5	Prof.Dr.dr. Celestyn Simanjongang Sp.PD	Jln. Rama Blok 12	04/07/1999	5564380548182

B. Diagram Perancangan Basis Data

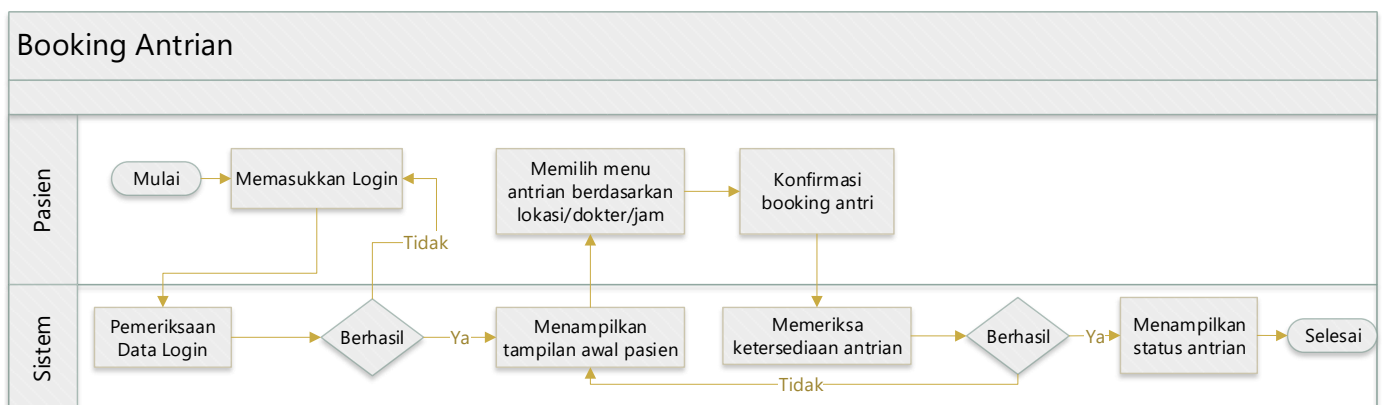
Berikut merupakan use case diagram yang ditunjukkan pada Gambar 1.



Gambar 1. Use Case Diagram.

Gambar 1. Merupakan diagram use case pada aplikasi mobile estimasi antrian, dengan tiga role/peran, yaitu dokter, admin, dan pasien, serta terdiri dari tujuh fitur.

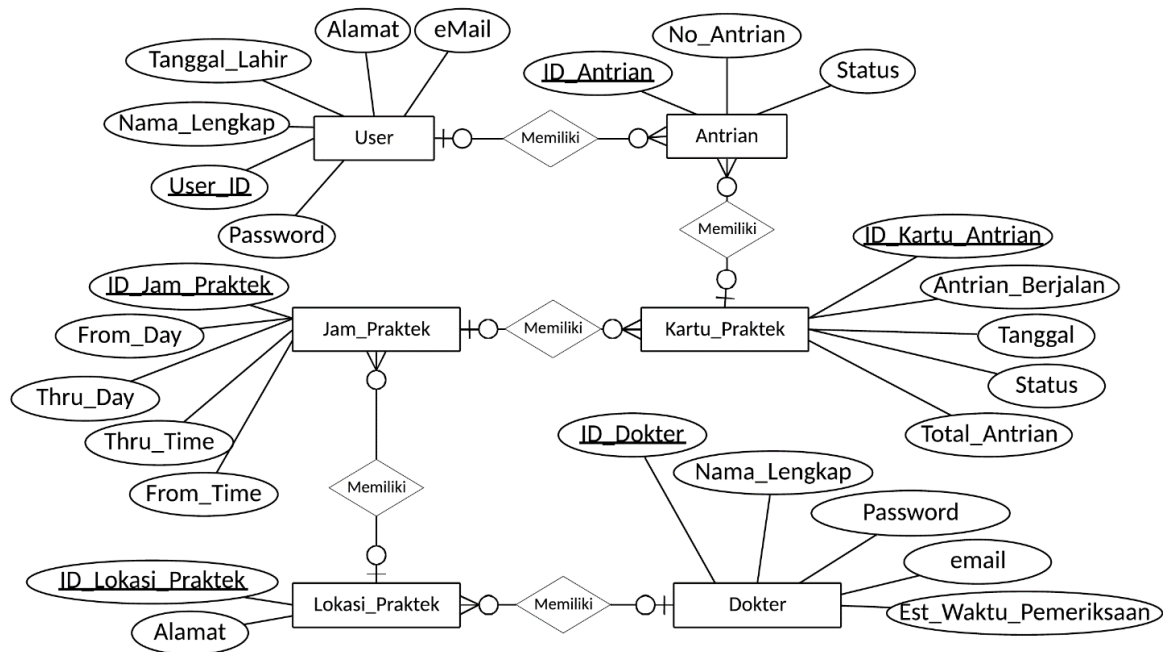
Berikut merupakan activity diagram dari use case aplikasi mobile estimasi antrian untuk fungsi mendaftarkan antrian oleh user/pasien yang diperlihatkan pada Gambar 2:



Gambar 2. Activity Diagram Booking Antrian.

Berikut merupakan rancangan penyimpanan basis data untuk aplikasi mobile estimasi antrian berupa ERD (Entity Relationship Diagram) yang digunakan untuk aplikasi mobile estimasi antrian, dengan 6 entitas yaitu User, Jam

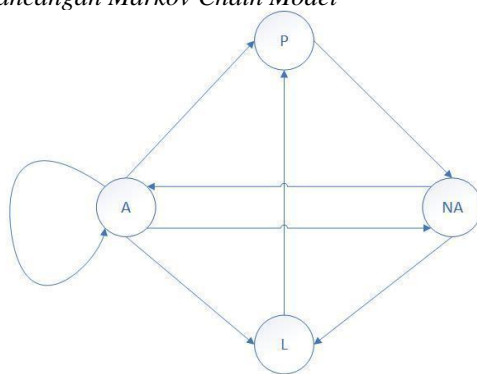
Praktek, Dokter, Antrian, Kartu Praktek, dan Lokasi Praktek yang ditunjukkan pada Gambar 3.



Gambar 3. Use Case Diagram.

Gambar 3. menunjukkan use case diagram yang digunakan pada aplikasi, dengan 6 entitas serta tanda panah (→) yang merupakan kardinalitas one, dan garis (-) yang merupakan many

C. Perancangan Markov Chain Model

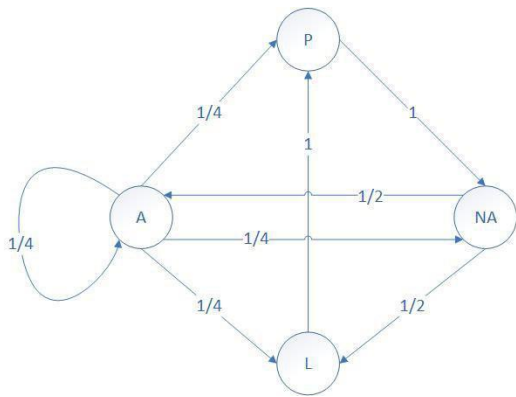


Gambar 4. Perancangan Model Markov Chain.

Gambar 4. menjelaskan model perancangan Markov Chain pada sistem estimasi waktu antrian yang belum memiliki bobot dengan keterangan sebagai berikut:

- A : State antrian.
- NA : State bukan antrian.
- L : Parameter tingkat kondisi darurat.
- P : State pemeriksaan oleh dokter.

Transisi dari state A ke P merupakan transisi yang berjalan sesuai dengan antrian normal. Jika terdapat kondisi yang memerlukan tindakan darurat atau yang memiliki tingkat kondisi darurat, maka penanganan akan lebih diutamakan dan akan langsung terjadi transisi dari state L ke P tanpa melalui state A. Proses dikatakan selesai jika tidak diperlukan penanganan lebih lanjut, dengan kata lain tidak perlu kembali lagi untuk diperiksa oleh dokter. Jika penanganan lebih lanjut seperti kembali lagi dalam tiga hari, atau perlu berobat jalan, maka transisi akan terjadi antara state P ke NA. Jika pasien memutuskan untuk berobat lebih lanjut maka akan didaftarkan ke antrian selanjutnya yang membuat terjadinya transisi antara state NA ke A. Jika pada state A terdapat antrian yang terlewati (skip), maka akan terjadi transisi dari state A ke A. Jika terdapat pasien yang memutuskan untuk tidak lanjut mengantri karena mungkin mendapatkan estimasi yang terlalu lama atau ingin pindah ke dokter lain, maka akan terjadi transisi dari state A ke NA. State L mempengaruhi semua state sebab state ini akan melakukan bypass state lain dan langsung masuk ke state penanganan.



Gambar 5. Perancangan Bobot Transisi Awal Antar State.

Untuk mendapatkan bobot PageRank, maka nilai transisi antar state disamaratakan. Nilai ini nantinya akan menjadi nilai awal untuk perhitungan bobot PageRank tersebut. Nilai transisi antar state dapat dilihat pada Gambar 5. Nilai transisi antar state diasumsikan sebagai variabel A dan dimasukkan ke dalam matriks

$$\begin{bmatrix} 1/4 & 0 & 1/2 & 0 \\ 1/4 & 0 & 0 & 1 \\ 1/4 & 1 & 0 & 0 \\ 1/4 & 0 & 1/2 & 0 \end{bmatrix}$$

Bobot awal setiap state akan diasumsikan sebagai variable v dengan nilai yang sama rata serta dimasukkan ke dalam matriks

$$\begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix}$$

Untuk mendapatkan bobot PageRank akhir, maka variable A akan dikalikan dengan variabel v sampai nilai yang dihasilkan tidak berubah ketika dikalikan kembali dengan variable A. Dalam kasus ini, nilai tersebut akan didapatkan ketika perkalian matriks telah dilakukan sebanyak 13 kali, atau  $A^{13}V$  sehingga akan didapatkan matriks sebagai berikut:

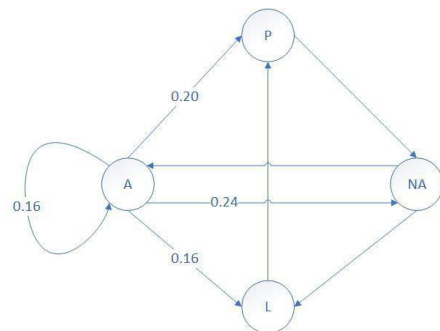
$$\begin{bmatrix} 0.16 \\ 0.20 \\ 0.24 \\ 0.16 \end{bmatrix}$$

Kemudian bobot matriks ini akan dimasukkan ke dalam Markov Chain Model sehingga didapatkan model akhir yang setiap state-nya sudah memiliki bobot PageRank, dapat dilihat pada Gambar 6.

$$A = \begin{bmatrix} 0.25 & 0 & 0.5 & 0 \\ 0.25 & 0 & 0 & 1 \\ 0.25 & 1 & 0 & 0 \\ 0.25 & 0 & 0.5 & 0 \end{bmatrix}$$

$$V = \begin{bmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{bmatrix}$$

$$\begin{aligned} AV &= \begin{bmatrix} 0.18 \\ 0.31 \\ 0.31 \\ 0.18 \end{bmatrix} & A^2V &= \begin{bmatrix} 0.19 \\ 0.22 \\ 0.35 \\ 0.19 \end{bmatrix} & A^3V &= \begin{bmatrix} 0.21 \\ 0.23 \\ 0.26 \\ 0.21 \end{bmatrix} & A^4V &= \begin{bmatrix} 0.18 \\ 0.26 \\ 0.28 \\ 0.18 \end{bmatrix} \\ A^5V &= \begin{bmatrix} 0.18 \\ 0.22 \\ 0.30 \\ 0.18 \end{bmatrix} & A^6V &= \begin{bmatrix} 0.19 \\ 0.22 \\ 0.26 \\ 0.19 \end{bmatrix} & A^7V &= \begin{bmatrix} 0.17 \\ 0.23 \\ 0.26 \\ 0.17 \end{bmatrix} & A^8V &= \begin{bmatrix} 0.17 \\ 0.23 \\ 0.26 \\ 0.17 \end{bmatrix} \\ A^9V &= \begin{bmatrix} 0.17 \\ 0.21 \\ 0.25 \\ 0.16 \end{bmatrix} & A^{10}V &= \begin{bmatrix} 0.16 \\ 0.21 \\ 0.25 \\ 0.16 \end{bmatrix} & A^{11}V &= \begin{bmatrix} 0.16 \\ 0.20 \\ 0.25 \\ 0.16 \end{bmatrix} \\ A^{12}V &= \begin{bmatrix} 0.16 \\ 0.20 \\ 0.24 \\ 0.16 \end{bmatrix} & A^{13}V &= \begin{bmatrix} 0.16 \\ 0.20 \\ 0.24 \\ 0.16 \end{bmatrix} \end{aligned}$$



Gambar 6. Markov Chain dengan bobot PageRank.

#### D. Model Estimasi Antrian

Dalam implementasinya, merujuk pada perhitungan PageRank, algoritma penghitungan estimasi setiap pasien dapat dilihat pada persamaan sebagai berikut:

```
const avg_waktu_pemeriksaan;
const total_antrian;

Integer antrian = nomor_antrian_user -
antrian_berjalan + 1;
Double estimasi_L = 0.16 * total_antrian;
Double estimasi_NA = 0.24 * total_antrian;
Double estimasi_AA = 0.16 * antrian;

Double estimasi_antrian_terakhir = (total_antrian
+ estimasi_L - estimasi_NA - estimasi_AA) *
avg_waktu_pemeriksaan;

Integer estimasi = Math.round((antrian /
total_antrian) * estimasi_antrian_terakhir);
```

Dari persamaan tersebut, dapat diperoleh perhitungan antrian adalah: asumsi setiap pasien diberikan waktu pemeriksaan selama 5 menit, lalu jumlah persentase pada total antrian di atas nomor antrian dihitung. Setelah itu, nilai

tersebut dikurangi dengan jumlah persentase pasien yang keluar dari antrian dan ditambah dengan waktu estimasi dari jumlah persentase pasien yang akan langsung diperiksa (jika dalam keadaan yang perlu didahulukan). Kemudian nilai total hasil perhitungan tersebut dijadikan estimasi lama antrian yang akan ditampilkan pada aplikasi.

**E. Arsitektur Sistem**

Berikut merupakan perubahan-perubahan rencana kerja yang terjadi selama pengembangan:

1. Penambahan konsep Markov Chain dan Algoritma PageRank.
2. Basis data menggunakan MongoDB.
3. Generator basis data menggunakan Python.
4. Penambahan Web Service REST API dengan menggunakan NodeJs.

**IV. HASIL EKSPERIMEN DAN EVALUASI**

**A. Tampilan Aplikasi**

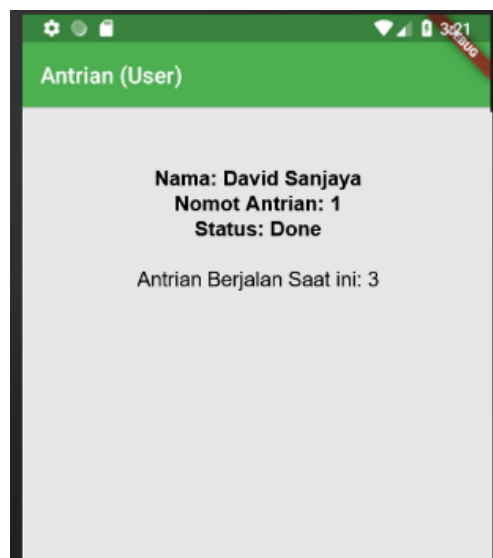
Berikut ini adalah tampilan antarmuka yang dibuat untuk aplikasi mobile sistem antrian:

1. Halaman update antrian merupakan halaman yang ditampilkan untuk dokter saat aplikasi antrian dibuka. Pada halaman update antrian terdapat menu / tombol update antrian. Gambar 7. merupakan implementasi rancangan antarmuka untuk tampilan update antrian.



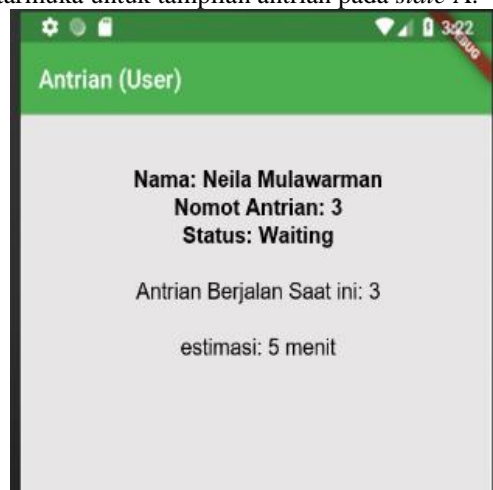
Gambar 7. Implementasi Tampilan Update Antrian.

2. Halaman antrian merupakan halaman yang ditampilkan untuk *user* saat aplikasi antrian dibuka. Gambar 8. merupakan implementasi rancangan antarmuka untuk tampilan antrian pada saat pasien / *user* setelah berada pada *state* P.



Gambar 8. Implementasi Tampilan Antrian Status Selesai ("Done").

3. Halaman antrian merupakan halaman yang ditampilkan untuk pasien saat aplikasi antrian dibuka. Gambar 9. merupakan implementasi rancangan antarmuka untuk tampilan antrian pada *state* A.



Gambar 9. Implementasi Tampilan Antrian Status Menunggu.

**B. Eksperimen**

Berikut ini merupakan eksperimen yang dilakukan dalam aplikasi antrian:

1. Perhitungan estimasi waktu dengan kondisi selisih nomor antrian dengan antrian berjalan adalah sama. Hasil eksperimen dapat dilihat pada Tabel IV. dan Tabel V.

TABEL IV  
TABEL DATA EKSPERIMEN PERTAMA (ANTRIAN 1)

Nomor Antrian	Total Antrian	Antrian Sekarang	Hasil Estimasi Waktu (Menit)
2	72	2	5
5	72	2	18

10	72	2	41
20	72	2	83
30	72	2	124
45	72	2	181

TABEL V  
TABEL DATA EKSPERIMEN PERTAMA (ANTRIAN 2)

Nomor Antrian	Total Antrian	Antrian Sekarang	Hasil Estimasi Waktu (Menit)
17	72	17	5
20	72	17	18
25	72	17	41
35	72	17	83
45	72	17	124
60	72	17	181

Dari Tabel IV. dan Tabel V. dapat dibuktikan bahwa hasil estimasi waktu yang didapatkan adalah sama jika selisih antara data “Nomor Antrian” dan “Antrian Sekarang” dari kedua tabel memiliki nilai yang sama, dengan asumsi nilai “Antrian Sekarang” tidak berubah (statis).

2. Perbandingan estimasi waktu antara dua nomor antrian yang berbeda (nomor antrian 20 dengan 40).

TABEL VI  
TABEL DATA EKSPERIMEN KEDUA (ANTRIAN 1)

Nomor Antrian	Total Antrian	Antrian Sekarang	Hasil Estimasi Waktu (Menit)
20	72	5	71
20	72	6	67
20	72	7	62
20	72	8	58
20	72	9	54
20	72	10	49

TABEL VII  
TABEL DATA EKSPERIMEN KEDUA (ANTRIAN 2)

Nomor Antrian	Total Antrian	Antrian Sekarang	Hasil Estimasi Waktu (Menit)
40	72	5	151
40	72	6	147
40	72	7	144
40	72	8	140
40	72	9	136
40	72	10	132

Dari Tabel VI. dan Tabel VII. dapat dibuktikan bahwa selisih hasil estimasi waktu untuk setiap penambahan jumlah antrian (1 nomor) akan semakin kecil jika nomor antrian semakin besar. Selisih hasil estimasi waktu antrian tidak berdasarkan pada “Nomor Antrian” dimana pola hasil estimasi tidak konstan.

3. Pengaruh Markov Chain terhadap estimasi waktu antrian

TABEL VIII  
TABEL DATA EKSPERIMEN KETIGA PERBANDINGAN ANTARA ESTIMASI WAKTU MENGGUNAKAN MARKOV DAN TANPA MARKOV

Nomor Antrian	Total Antrian	Antrian Sekarang	Hasil Estimasi Waktu (Menit)	Hasil Estimasi Waktu Tanpa Markov
20	72	5	71	80
20	72	6	67	75
20	72	7	62	70
20	72	8	58	65
20	72	9	34	60
20	72	10	49	55

Dari Tabel VIII. dapat dibuktikan bahwa:

1. Hasil perhitungan estimasi waktu dengan menggunakan model Markov Chain memberikan nilai yang lebih kecil.
2. Semakin kecil selisih antara “Nomor Antrian” dengan “Antrian Sekarang”, maka akan semakin kecil selisih antara estimasi waktu perhitungan tanpa Markov dengan perhitungan menggunakan Markov.
3. Selisih estimasi waktu untuk setiap kenaikan 1 nomor dengan menggunakan Markov Chain tidak selalu konstan.

### C. Flutter Testing

Pada testing aplikasi yang dikembangkan, tools yang digunakan adalah Flutter Testing, yang merupakan unit testing yang telah disediakan flutter untuk melakukan Test Driven Development. Contoh penggunaan Flutter Testing dapat dilihat pada persamaan sebagai berikut:

```
test('return a user\'s data if the http call
completes successfully',
  () async {
    final client = MockClient();

    when(client.get(
      argThat(
        startsWith('http://localhost:3000/antrian/user/5d0
891cd1664c04614648b27'),
      ),
      headers: anyNamed('headers'),
    ),).thenAnswer((_) async =>
      http.Response('{"antrian": "{}"}', 200));

    expect(await getAntrian(client),
      isInstanceOf<UserDetail>());
  });
```

Kode tersebut adalah contoh unit testing untuk memeriksa koneksi antara aplikasi dengan web server. Hasil testing yang berhasil, menunjukkan bahwa aplikasi dan web server terhubung



## V. KESIMPULAN DAN SARAN

Dari aplikasi *mobile* untuk sistem antrian praktek dokter yang dilengkapi dengan perhitungan estimasi waktu, dilakukan tiga buah percobaan yaitu perhitungan estimasi waktu berdasarkan selisih nomor antrian yang sama, perbandingan estimasi waktu antara dua nomor antrian yang berbeda, dan perbandingan estimasi waktu antara sistem yang mengimplementasikan Markov Chain dengan sistem tanpa Markov Chain. Dalam penelitian ini tidak diperhitungkan estimasi waktu pada saat pasien diperiksa oleh dokter, sehingga tidak ada *hidden state* pada model Markov. Dari hasil perhitungan bobot estimasi waktu antrian menggunakan konsep Markov Model dengan algoritma PageRank, hasil estimasi waktu akan sama jika selisih antrian adalah sama (Tabel IV. dan Tabel V.). Akan tetapi semakin besar selisih nomor antrian dengan antrian berjalan, maka perubahan estimasi waktu untuk setiap kenaikan nomor antrian berjalan akan semakin kecil (Tabel VI. dan Tabel VII.). Estimasi waktu dengan memperhitungkan faktor acak yang mempengaruhi antrian dapat diandalkan, seperti ditunjukkan pada Tabel VIII. yang membandingkan hasil estimasi waktu antrian menggunakan Markov dengan hasil estimasi waktu antrian linier. Akhirnya, aplikasi *mobile* untuk sistem antrian yang mengimplementasikan konsep Markov Chain dengan perhitungan estimasi waktu menggunakan algoritma PageRank dapat didemonstrasikan untuk menunjukkan perhitungan estimasi waktu antrian yang lebih mendekati kenyataan dengan memperhitungkan faktor acak.

## UCAPAN TERIMA KASIH

Terima kasih diucapkan sebesar – besarnya kepada para pengajar dan staf Fakultas Teknologi Informasi, Prodi Magister Ilmu Komputer Universitas Kristen Maranatha

Bandung atas dukungan dan bantuannya dalam penyusunan laporan penelitian ini

## DAFTAR PUSTAKA

- [1] B. A. Ahmad, K. Khairatul, & A. Farnaza, *An Assessment of Patient Waiting and Consultation Time in a Primary Healthcare Clinic*, 2017.
- [2] A. A. Onoja & M. Kembe, *The Application of Queuing Theory on Patient Waiting Time in Ante-natal Care Clinic*, *The International Journal of Science & Technology*, 2018.
- [3] E. Fosler-Lussier, *Markov Models and Hidden Markov Models: A Brief Tutorial*, *International Computer Science Institute*, December 1998.
- [4] P. Rai, M. G. Meitei, F. Khatun, A. Choudhury, U. Kr. Chakraborty, *Commentary on Application of Hidden Markov Model in Google Page Ranking*, *International Journal of Computer Applications, International Conference on Computing & Communication*, 2016.
- [5] M.E. Prasetyo, *Teori Dasar Hidden Markov Model*, Bandung, 2010.
- [6] E. Renaldy, *Implementasi Algoritma PageRank Pada Pemrograman Berorientasi Objek (OOP) (Studi kasus Menggunakan Bahasa Pemrograman Java)*, 2012.
- [7] M. Luqman, *Keamanan Perangkat Lunak Pada Bahasa Pemrograman Node.JS untuk Aplikasi Berbasis Web*, 2016.
- [8] D. Lin, J. Patrick, & F. Labeau, *Estimating the waiting time of multi-priority emergency patients with downstream blocking*, *Health Care Management Science*, vol. 17, issue 1, pp 88-99, 2014.
- [9] Zulfikar & R. Arif, *Rancang Bangun Aplikasi Antrian Poliklinik Berbasis Mobile*, *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 5, issue 3, 2018.
- [10] A. N. Langville & C. D. Meyer, *Updating Markov Chains with an Eye on Google's PageRank*, *SIAM Journal on Matrix Analysis and Application*, vol. 27, issue 4, pp 968-987, 2006.
- [11] K. Kalecky & Y. R. Cho, *PrimAlign: PageRank-inspired Markovian alignment for large biological networks*, *Bioinformatics*, vol. 34, issue 13, pp i537-i546, 2018.
- [12] (2019) *PageRank Algorithm*. [Online]. Tersedia: <http://pi.math.cornell.edu/~mec/Winter2009/RalucaRemus/Lecture3/lecture3.html>
- [13] (2019) *SearchApp Architecture Website*. [Online]. Tersedia: <https://searchmicroservices.techtarget.com/tip/REST-vs-SOAP-Choosing-the-best-web-service/>