

Rancang Bangun Aplikasi Deteksi Dini Kanker Ovarium Berbasis Android Dengan *Expert System*

1st Muhammad Zuhairi
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

zuhairiui@student.telkomuniversity.ac.id

2nd Hilman Fauzi Tresna Sania Putra
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

hilmanfauzi@telkomuniversity.ac.id

3rd Favian Dewanta
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

favian@telkomuniversity.ac.id

Abstrak—Kanker ovarium adalah tumor yang timbul pada rahim dan dapat berakibat fatal apabila tidak ditangani sedini mungkin. Kanker ovarium sulit terdeteksi pada stadium awal karena tidak menimbulkan gejala spesifik dan umumnya baru terdeteksi pada stadium lanjut. Pada stadium lanjut kanker tersebut akan lebih sulit untuk ditangani dan biaya pengobatannya pun menjadi lebih mahal. Oleh karena itu dibuatlah penelitian dengan membuat aplikasi android untuk mendeteksi kanker ovarium sedini mungkin dengan metode sistem pakar. Diperoleh data kuesioner pasien kanker dari Rumah Sakit Hasan Sadikin Bandung yang akan menjadi acuan dalam pembuatan aplikasi. Penulis membandingkan keakuratan prediksi data menggunakan algoritma K-Nearest Neighbor, Support Vector Machine, dan Random Forest. Lalu dilanjutkan dengan pembuatan aplikasi serta dilakukan analisis *Quality of Services* menggunakan Apache JMeter. Hasil pengujian menunjukkan bahwa algoritma K-Nearest Neighbor menggunakan nilai $K=7$ mendapatkan nilai akurasi sebesar 92.3%. Maka dari itu, dipilihlah algoritma K-Nearest Neighbor untuk diimplementasikan ke dalam aplikasi deteksi kanker ovarium. Dengan memasukkan algoritma K-Nearest Neighbor ke dalam aplikasi, didapatkan hasil akurasi pada aplikasi sebesar 80,23%. Lalu analisis *Quality of Service* menghasilkan *throughput* terbaik sebesar 2.89 kbps, dan *latency* terbaik sebesar 13s, masing-masing dengan beban 100 user dan mampu menangani permintaan dari 100 dan 150 user tanpa error.

Kata Kunci: *Android, K-Nearest Neighbor, Kanker Ovarium, Random Forest, Sistem Pakar, Support Vector Machine.*

I. PENDAHULUAN

Kanker ovarium adalah tumor ganas pada ovarium yang dapat menyerang wanita dari segala usia, terutama wanita yang telah lanjut usia dan wanita yang memiliki keluarga dengan riwayat kanker ovarium [1].

Kanker ovarium sulit terdeteksi pada stadium awal, kebanyakan baru diketahui setelah memasuki stadium lanjut dan akhir. Ketika memasuki tahap stadium lanjut dan akhir, biaya pengobatan akan menjadi semakin mahal dan kanker tersebut akan lebih sulit diobati [2]. Oleh karena itu, penting bagi perempuan untuk melakukan pemeriksaan berkala ke tenaga kesehatan agar dapat dilakukan penanganan yang

tepat apabila terindikasi mengidap kanker ovarium. Dari permasalahan tersebut maka diperlukan sebuah metode yang dapat melakukan sebuah deteksi dini terhadap gejala yang ada.

Tugas Akhir ini bertujuan untuk merancang sistem machine learning yang dapat mendeteksi kanker ovarium dan mengimplementasikannya pada aplikasi android. Aplikasi tersebut, diharapkan dapat membantu meningkatkan kesadaran masyarakat tentang kanker ovarium dan solusi penanganannya. Pada penelitian ini, penulis menggunakan metode K-Nearest Neighbors (KNN), *Support Vector Machine* (SVM), dan *Random Forest* untuk memprediksi penyakit kanker ovarium dari gejala yang ada. Penulis melakukan perbandingan hasil performansi dari ketiga metode tersebut, setelah didapatkan hasil terbaik dari ketiga algoritma tersebut, model *machine learning* yang sudah dibuat akan diimplementasikan pada aplikasi android.

Pada tahun 2021, Lastri Kartika Sari dalam penelitiannya menggunakan machine learning dengan metode *Decision Tree* untuk mendeteksi kanker ovarium [3]. Tingkat akurasi yang dapat diberikan oleh sistem *machine learning* dengan menggunakan *Decision Tree* berada di angka 77.27%. Kemudian sistem tersebut diimplementasikan ke dalam aplikasi android dan didapatkan hasil akurasi sebesar 65.45%.

Lalu penulis mengusulkan untuk menggunakan metode yang biasa digunakan untuk pengklasifikasian, yaitu KNN, SVM, dan *Random Forest*. Metode KNN diusulkan karena metode ini merupakan teknik paling dasar dan sederhana untuk melakukan klasifikasi. KNN juga memiliki beberapa keunggulan seperti pelatihan yang sangat cepat dan sederhana serta efektif untuk pelatihan dengan jumlah data yang banyak. Namun metode KNN memiliki kekurangan seperti nilai K yang kita tidak tahu mana nilai K yang paling bagus. Sementara itu, SVM diusulkan karena metode ini memiliki kelebihan seperti dapat melakukan klasifikasi data yang terpisah secara linier (*linearly separable*) dan non-linier

(*nonlinear separable*) Lalu, *Random Forest* diusulkan karena merupakan lanjutan dari algoritma *Decision Tree* yang dipakai pada penelitian sebelumnya, *Random Forest* mampu mengklasifikasikan data yang memiliki jumlah sampel yang banyak.

II. KAJIAN TEORI

A. Kanker Ovarium

Kanker ovarium timbul pada jaringan indung telur, yaitu dua organ yang berada di sisi kanan dan kiri rahim dan lebih sering terjadi pada wanita berusia menengah maupun wanita lanjut usia yang telah mengalami pascamenopause. Kanker ini menempati urutan ketiga kanker ginekologi paling sering terjadi dan memiliki prognosis terburuk serta tingkat kematian tertinggi [4]. Pada tahap awal, kanker ovarium tidak menimbulkan gejala spesifik dan biasanya baru terdeteksi ketika mencapai stadium lanjut. Ini menjadi salah satu penyebab tingginya tingkat kematian akibat kanker ovarium. Gambar 1 merupakan bentuk kanker ovarium yang berada pada sel indung telur.



GAMBAR 1

KANKER OVARIUM PADA SEL INDUNG TELUR.

Pada stadium awal, penderita kanker ovarium tidak memiliki gejala sampai kanker tersebut menyebar di daerah panggul dan perut. Ketika memasuki stadium lanjut, penderita hanya memiliki beberapa gejala penyakit umum seperti hilangnya nafsu makan dan penurunan berat badan. Oleh karena itu kanker ovarium sangat sulit terdeteksi ketika memasuki stadium awal dan lanjut, dan ketika memasuki stadium akhir kanker tersebut akan lebih sulit diobati dan bisa berakibat fatal.

B. Sistem Pakar

Sistem pakar adalah suatu program komputer atau sistem informasi yang mengandung beberapa pengetahuan dari pakar terkait suatu bidang yang spesifik [5]. Pakar yang dimaksudkan merupakan seseorang yang memiliki keahlian khusus, contohnya dokter, psikolog, mekanik, dan lain sebagainya. Sistem pakar memiliki kemampuan untuk merekomendasikan rangkaian tindakan pengguna untuk dapat menjalankan sistem koreksi yang tepat dan akurat. Dalam pengembangannya, sistem pakar tersusun atas beberapa komponen atau struktur pembentuk sebuah sistem informasi yang komprehensif di antaranya adalah :

a. User Interface

Antarmuka atau interface merupakan mekanisme yang digunakan sebagai sarana untuk berkomunikasi dan berinteraksi.

b. Knowledge Base

Knowledge Base atau basis pengetahuan mengandung pemahaman mengenai formulasi dan skema penyelesaian masalah.

c. Interface Engine

Komponen ini mengandung seperangkat algoritma, yang bertujuan untuk melakukan penalaran, penilaian dan pengambilan keputusan untuk dapat memecahkan suatu masalah dengan baik dan nantinya akan diolah menjadi sebuah kesimpulan.

C. Machine Learning

Machine Learning adalah bidang keilmuan yang bertujuan untuk memahami atau mengenali struktur suatu data dan mengonversi data tersebut pada suatu model. Pada umumnya, algoritma *machine learning* dibuat untuk menangani tiga kebutuhan, yaitu klasifikasi, prediksi, dan *clustering*. Klasifikasi sendiri adalah suatu kegiatan untuk mengelompokkan data berdasarkan kategori tertentu yang akan digunakan dalam mendeteksi suatu data baru. Prediksi adalah suatu kegiatan dalam menebak nilai output berdasarkan nilai input yang ditentukan. *Clustering* adalah kegiatan dalam mengelompokkan data berdasarkan tingkat kemiripan data [6]. Contoh dari *machine learning* adalah prediksi kata yang ada di *smartphone* kita atau pendeteksian penyakit seseorang dari gejala yang ada. Hal tersebut bisa dilakukan karena program pada kedua hal tersebut telah menyusun pengetahuan dari data yang ada, biasanya dalam bentuk sebuah model matematis. *Machine learning* tidak akan bisa bekerja tanpa adanya data. Karena pada dasarnya istilah *machine learning* adalah proses komputer untuk mempelajari data [7]. Data biasanya akan dibagi menjadi 2 kelompok, yaitu data *training* dan data *testing*. Data *training* nantinya akan digunakan untuk melatih algoritma untuk mencari model yang cocok, sementara data *testing* akan dipakai untuk mengetes dan mengetahui performa model yang didapatkan pada tahapan *testing*.

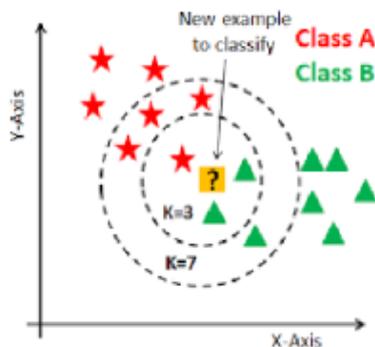
D. Algoritma Machine Learning

Machine learning memerlukan algoritma untuk dapat digunakan, ada beberapa algoritma Machine Learning yang dapat digunakan di antaranya adalah K-Nearest Neighbor, *Support Vector Machine*, dan *Random Forest* [8].

a. K-Nearest Neighbor (KNN)

KNN adalah metode klasifikasi yang dapat melakukan prediksi terhadap suatu objek berdasarkan data latih yang jaraknya paling dekat dengan objek tersebut. Algoritma KNN

termasuk dalam golongan supervised learning yang bertujuan untuk menemukan pola baru dalam data dengan menghubungkan pola data yang sudah ada dengan data yang baru. Pada algoritma KNN data akan dibagi menjadi data latih dan data uji. Data latih akan diproyeksikan ke ruang berdimensi banyak, di mana masing-masing dimensi merepresentasikan fitur dari data [9]. Ilustrasi KNN dapat dilihat pada Gambar 2.



GAMBAR 2
ILUSTRASI ALGORITMA KNN

Data uji diubah menjadi sebuah titik yang berada di dalam sebuah ruang dimensi. Titik tersebut akan dimasukkan ke dalam sebuah kelas tertentu dan akan dibandingkan dengan semua titik yang ada di dalam data latih. Setelah itu akan dicari nilai K yang paling mirip dengan titik tersebut. KNN memiliki beberapa formula, di antaranya adalah *Euclidean Distance*, *Manhattan Distance*, *Hamming Distance*, dan *Minkowski Distance*. Penulis menggunakan formula *Euclidean Distance* pada penelitian ini. Formula *Euclidean Distance* dapat dilihat pada Rumus (1) di bawah ini.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{1}$$

dimana:

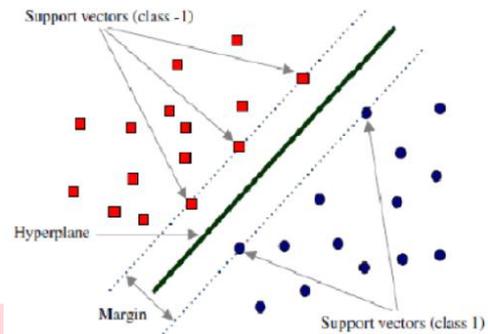
- d = jarak
- x_1 = data latih
- x_2 = data uji
- y_1, y_2 = fitur lainnya

Data uji diubah menjadi sebuah titik yang berada di dalam sebuah ruang dimensi. Titik tersebut akan dimasukkan ke dalam sebuah kelas tertentu dan akan dibandingkan dengan semua titik yang ada di dalam data latih. Setelah itu akan dicari nilai K yang paling mirip dengan titik tersebut.

b. *Support Vector Machine (SVM)*

(SVM) adalah sebuah algoritma klasifikasi yang dikembangkan oleh Vapnik dan rekan-rekannya di Laboratorium Bell. Algoritma SVM memiliki prinsip dasar linear classifier yang dapat memisahkan kasus klasifikasi secara linear. Algoritma ini bertugas untuk mendefinisikan

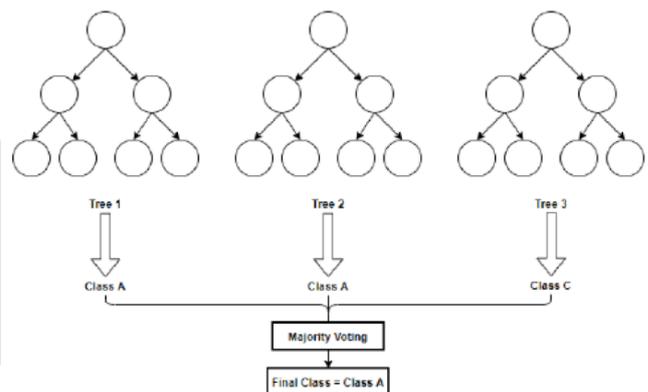
garis pemisah atau *hyperplane* [10]. Tingkat ideal dari *hyperplane* ditentukan dari jarak margin *hyperplane* dengan titik terdekat dari masing-masing kelas. Ilustrasi algoritma SVM dapat dilihat pada Gambar 3.



GAMBAR 3
ILUSTRASI ALGORITMA SVM

c. *Random Forest*

Random Forest menggunakan sebuah metode untuk meningkatkan akurasi klasifikasi dengan cara mengkombinasikan metode klasifikasi atau yang biasa disebut *ensemble*. *Random Forest* terdiri dari kumpulan *Decision Tree* yang digunakan untuk mengklasifikasi data ke suatu kelas. *Random Forest* termasuk ke dalam algoritma *Supervised*. Metode ini menciptakan hutan (forest) dengan sejumlah pohon (*tree*) sebanyak yang kita mau [11]. Lalu hasil kelas dari pohon-pohon tersebut akan dilakukan *voting* berdasarkan kelas terbanyak. Ilustrasi *Random Forest* dapat dilihat pada Gambar 4.



GAMBAR 4
ILUSTRASI ALGORITMA RANDOM FOREST

E. *Flask*

Flask merupakan suatu framework *microservice* berbasis bahasa *python*. Dengan menggunakan *flask*, proses pengintegrasian *machine learning* menggunakan *library python* akan menjadi mudah karena *flask* sendiri sudah satu platform dengan *python*. *Flask* dirasa tepat untuk dipakai dalam penelitian ini, karena pada penelitian ini penulis hanya menggunakan sebuah *service* sederhana untuk mengkomputasi

data. *Flask* tidak memiliki banyak *tools* dan *library* sehingga dapat digunakan untuk menghemat waktu dalam melakukan pembangunan web meskipun dengan *library* yang sederhana [12].

III. METODE

Pada bagian ini akan dijelaskan alur dari perancangan sistem machine learning, aplikasi android, dataset yang digunakan, dan pengujian performansi sistem machine learning serta aplikasi android.

A. Alur Sistem Perancangan

Pada bagian ini akan dibahas perancangan sistem yang digunakan untuk mendiagnosa terdeteksinya kanker ovarium, alur sistem dapat dilihat pada Gambar 5.



GAMBAR 5
BLOK DIAGRAM ALUR SISTEM

Blok diagram di atas merupakan representasi sistem pada penelitian ini. Pertama dilakukan akuisisi data, data didapatkan dari hasil kuesioner pasien penderita kanker di Rumah Sakit Hasan Sadikin Bandung. Data tersebut akan diolah dan hasilnya akan digunakan dalam pembuatan model sistem yang bertujuan untuk mengklasifikasikan data tersebut menjadi "beresiko kanker" atau "tidak beresiko kanker", lalu model tersebut akan diimplementasikan pada aplikasi deteksi dini kanker ovarium.

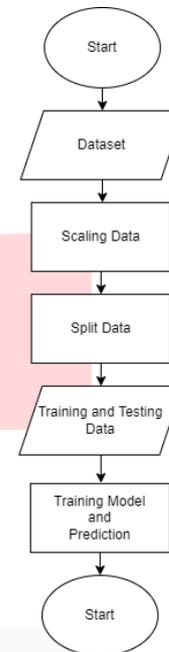
B. Akuisisi Data

Akuisisi data merupakan proses mendapatkan data dari permasalahan yang akan dibahas pada penelitian ini. Penulis bekerja sama dengan Rumah Sakit Hasan Sadikin Bandung untuk melakukan pengambilan data mengenai kanker ovarium. Data yang diambil merupakan data hasil kuesioner pasien yang terdiagnosa penyakit kanker ovarium. Dari hasil diagnosa tersebut akan dijadikan beberapa pertanyaan seperti apakah terjadi gangguan menstruasi, perut terasa membesar, nyeri perut, perut kembung, mual/muntah, nafsu makan menurun, gangguan buang air kecil dan besar, cepat kenyang dan penurunan berat badan. Didapatkan sebanyak 172 data CSV yang dibagi menjadi 52 data uji dan 120 data latih.

C. Pembuatan Model *Machine Learning*

Pada penelitian ini dilakukan pembuatan model untuk memprediksi data pasien yang beresiko dan tidak beresiko kanker. Pembuatan model dilakukan menggunakan tiga model *machine learning* yaitu KNN, SVM dan *Random Forest*. Pada model KNN digunakan *K Generator* untuk mencari nilai K terbaik. *K Generator* bekerja dengan cara melooping nilai K dari satu sampai jumlah data test dan dihitung masing-masing

akurasinya. Lalu pada model SVM digunakan kernel linear karena merupakan fungsi kernel yang sederhana dan cocok ketika terdapat banyak fitur pada data. Pada model random forest digunakan sebanyak 100 *n estimators* yang artinya model tersebut membuat 100 *decision tree*. *Flow chart* model *machine learning* dapat dilihat pada Gambar 6.



GAMBAR 6
FLOWCHART MODEL MACHINE LEARNING

Pertama-tama dataset yang didapatkan dari Rumah Sakit Hasan Sadikin Bandung berupa *file* CSV akan dimasukkan, lalu data tersebut akan dinormalisasi pada proses scaling data. Proses normalisasi atau scaling bertujuan untuk membuat beberapa fitur yang ada pada dataset memiliki rentang nilai yang sama sehingga dapat membuat analisis statistik menjadi lebih mudah. Proses normalisasi pada model machine learning ini menggunakan metode *standard score*. Normalisasi menggunakan metode *standard score* akan menghasilkan nilai baru yang berkisar pada angka 0 dan biasanya ada pada rentang antara -3 dan 3. Setelah dilakukan normalisasi, data akan dipisah menjadi data *training* dan data *test*, kedua data tersebut akan dimasukkan ke dalam *training* model dan dilakukan proses prediksi.

Setelah itu dilakukan uji performansi pada tiga model tersebut menggunakan beberapa parameter, di antaranya adalah akurasi, *error rate*, spesifisitas, presisi, *recall* dan *f-1 score*. Uji performansi dengan parameter tersebut dapat dilakukan menggunakan *confusion matrix*.

D. *Confusion Matrix*

Confusion Matrix berfungsi untuk memberikan informasi perbandingan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi sebenarnya [13]. *Confusion matrix* digunakan untuk menghitung parameter performansi

seperti akurasi, *error rate*, spesifisitas, presisi, *recall* dan *f-1 score*. Gambar 7 merupakan tabel *confusion matrix*.

Nilai Aktual	Positif	TP	FN
	Negatif	FP	TN
		Positif	Negatif
		Nilai Prediksi	

GAMBAR 7
CONFUSION MATRIX

Dimana :

- True Positive* (TP) merupakan jumlah data bernilai positif dan terprediksi benar sebagai nilai positif. Sebagai contoh, jumlah label "BERESIKO" yang terdeteksi dengan benar.
- False Positive* (FP) merupakan jumlah data yang bernilai negatif namun terprediksi sebagai positif. Sebagai contoh, jumlah label "TIDAK BERESIKO" terprediksi salah, sehingga dinyatakan sebagai label "BERESIKO".
- True Negative* (TN) merupakan jumlah data yang bernilai negatif dan terprediksi benar sebagai data negatif. Contohnya adalah, jumlah label "TIDAK BERESIKO" yang terprediksi benar.
- False Negative* (FN) merupakan jumlah data yang bernilai positif namun terprediksi sebagai data negatif, Contohnya adalah, jumlah label "BERESIKO" yang salah terprediksi sehingga dinyatakan "TIDAK BERESIKO".

Dari pengertian tersebut dapat disimpulkan bahwa prediksi benar dinyatakan dengan *true* dan jika prediksi salah dinyatakan dengan *false*. Dari Gambar 7 dapat diukur parameter perfomansi sistem seperti akurasi, *error rate*, spesifisitas, presisi, *recall* dan *f-1 score*. Berikut adalah persamaan-persamaan parameter yang dapat dihitung menggunakan *confusion matrix*.

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

$$ErrorRate = \frac{FP + FN}{TP + TN + FP + FN} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$Spesifitas = \frac{TN}{TN + FP} \tag{5}$$

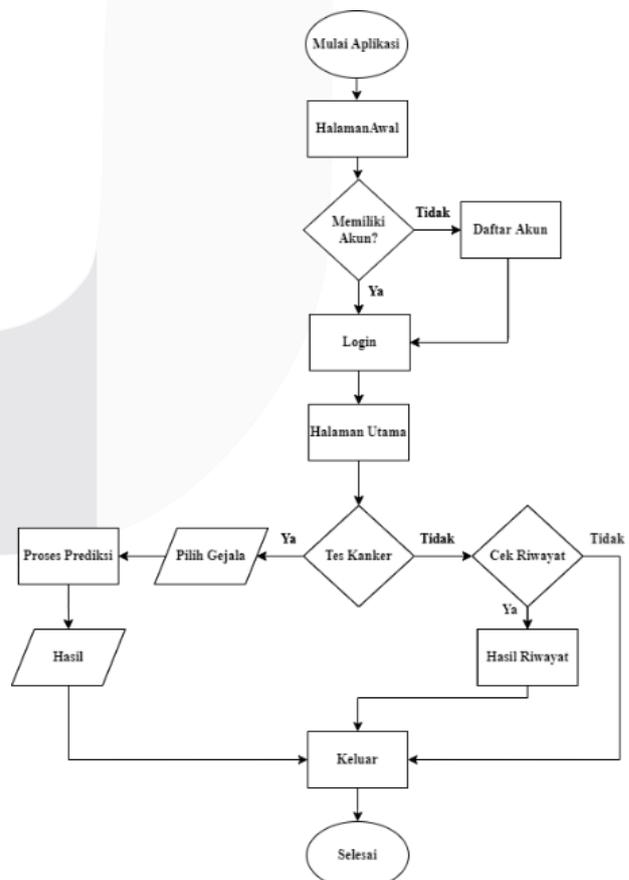
$$Presisi = \frac{TP}{TP + FP} \tag{6}$$

$$f - 1 \text{ score} = 2 \times \frac{Recall \times Presisi}{Recall + Presisi} \tag{7}$$

Akurasi merupakan probabilitas dari data yang terprediksi dengan benar. Akurasi dinyatakan sebagai rasio prediksi antara *true positive* dan *true negative*. *Error rate* atau *loss* adalah ketidaktepatan sistem dalam mendeteksi input, sehingga *output* yang dihasilkan tidak sesuai dengan apa yang diharapkan. *Recall* bertujuan untuk mengukur seberapa baik kemampuan sistem dalam mendeteksi data *positive* [14]. Spesifisitas berfungsi untuk memperkirakan seberapa besar kemungkinan data uji negative dapat diprediksi dengan benar. Presisi merupakan tingkat akurasi dalam memprediksi data *negative* yang sesuai. *f-1 score* berfungsi untuk menyeimbangkan nilai *recall* dan presisi dengan membandingkan rata-rata nilai *recall* dan presisi yang telah didapatkan.

E. Pembuatan Aplikasi

Penulis akan membuat aplikasi berbasis android agar dapat digunakan dan dilakukan simulasi untuk mendapatkan hasil dari pengujian yang akan dianalisis. Akan dibuat perancangan sistem aplikasi android seperti pada Gambar 8.



GAMBAR 8
FLOWCHART APLIKASI ANDROID

Pada Gambar 8 menunjukkan saat aplikasi dibuka akan menampilkan halaman awal. Lalu *user* akan diarahkan pada halaman *login*, pada halaman *login* user perlu melakukan proses autentikasi dengan menginput *email* dan *password*. Apabila *user* belum memiliki akun, *user* akan diarahkan untuk mendaftarkan akun dan akan diarahkan ke halaman *login* kembali untuk proses autentikasi. Apabila autentikasi benar maka *user* akan masuk ke halaman utama. Selanjutnya *user* akan memilih menu test kanker ovarium dan akan memilih gejala pada halaman pilih gejala. Setelah selesai memilih gejala, sistem akan melakukan pengujian berdasarkan inputan gejala dari *user* dan setelah itu aplikasi akan menampilkan hasil dari test kanker ovarium. *User* dapat memilih fitur cek riwayat apabila *user* ingin melihat riwayat tes kanker ovarium. Selanjutnya apabila *user* telah selesai menggunakan aplikasi, *user* dapat keluar dari aplikasi dengan mengklik tombol *logout*.

F. Pengujian Akurasi Aplikasi

Pengujian nilai akurasi dilakukan untuk mengetahui seberapa baik aplikasi dalam mendeteksi kanker ovarium. Cara untuk menghitung tingkat akurasi sistem dapat dilakukan dengan membandingkan jumlah data benar dengan jumlah data uji. Rumus yang digunakan untuk menghitung tingkat keakuratan sistem dapat dilihat pada persamaan ... Total data benar merupakan jumlah dari data yang terprediksi dengan benar sesuai dengan data asli, sedangkan total data uji merupakan seluruh data asli.

$$\text{Akurasi (\%)} = \frac{\text{total data benar}}{\text{total data}} \times 100\% \quad (8)$$

G. Pengujian *Quality of Service*

Pengujian *quality of service* akan dilakukan menggunakan aplikasi Apache JMeter dengan melihat beberapa parameter sebagai berikut :

a. *Latency*

Pengukuran *latency* bertujuan untuk melihat waktu yang dibutuhkan permintaan mencapai ke server dan menanggapi perjalanan balik. Rumus yang digunakan pada *latency* yaitu:

$$\text{Latency} = \frac{\text{Jumlah Nilai Latency}}{\text{Jumlah Sample}} \quad (9)$$

Dimana jumlah nilai *latency* merupakan jumlah seluruh waktu yang dibutuhkan *request* menuju *server*. Sedangkan jumlah *sample* adalah jumlah *user* yang melakukan *request* ke *server*. Satuan yang dipakai dalam *latency* adalah *second*.

b. *Throughput*

Pengukuran *throughput* bertujuan untuk menghitung jumlah permintaan permenit yang diproses oleh *server*.

Rumus yang digunakan untuk menghitung *throughput* adalah:

$$\text{Throughput} = \frac{\text{Ukuran Data}}{\text{Waktu yang Dibutuhkan}} \quad (10)$$

Dimana ukuran data merupakan jumlah *user* dalam melakukan *request* ke *server* dan waktu yang dibutuhkan adalah waktu yang dibutuhkan bagi *server* mengirimkan data ke *user*. Satuan yang dipakai dalam *throughput* adalah Kbps.

c. *Load Performance*

Load performance bertujuan untuk mengetahui kemampuan dari server dalam menanggapi *request* dari *user*.

IV. HASIL DAN PEMBAHASAN

Pada bab ini penulis akan membahas tentang hasil dan analisis dari sistem serta aplikasi deteksi dini kanker ovarium. Terdapat lima pengujian yang akan dibahas yaitu :

1. Pengujian performansi model *machine learning*, pengujian ini bertujuan untuk menentukan hasil performansi terbaik dari tiga algoritma *machine learning*, yaitu algoritma KNN, SVM, *Random Forest* yang akan diimplementasikan pada aplikasi android.
2. Pengujian performansi aplikasi android, bertujuan untuk mengetahui hasil akurasi pada aplikasi android.
3. Pengujian *quality of service*, bertujuan untuk mengetahui *quality of service* pada sistem yang dideploy pada *cloud server*.
4. Pengujian *black box*, bertujuan untuk mengetahui apakah fungsi software tersebut dapat berfungsi dengan baik.
5. Pengujian CPU *usage* dan *memory usage*.

A. Pengujian Performansi dan Analisis Model *Machine Learning*

Pengujian ini bertujuan untuk mengetahui hasil performansi dari model *machine learning* yang telah dibuat. Dilakukan pemisahan data pada 172 data yang telah didapatkan, data tersebut dibagi menjadi dua yaitu data uji dan data latih. rasio pembagian data yang digunakan yaitu 30:70, di mana 30% untuk data uji dan 70% untuk data latih. Pada pengujian model KNN, digunakan K generator untuk mengetahui nilai K yang terbaik dan didapatkan nilai K = 7. Lalu untuk pengujian model SVM, digunakan kernel linear, kernel linear merupakan fungsi kernel yang paling sederhana dan cocok ketika terdapat banyak fitur pada data. Dan yang terakhir pada pengujian random forest digunakan n estimators = 100 yang artinya model membuat 100 decision tree. Pengujian performansi model *machine learning* dilakukan menggunakan confusion matrix. Hasil confusion matrix dapat dilihat pada Tabel 1.

TABEL 1

HASIL CONFUSION MATRIX

Machine Learning Algorithm	True Positive	True Negative	False Positive	False Negative
K-Nearest Neighbor	24	24	2	2
Random Forest	24	23	3	2
Support Vector Machine	22	16	10	4

Setelah didapatkan hasil *confusion matrix*, Hasil performansi pada model *machine learning* yang dibuat dapat diketahui. Hasil performansi pada model *machine learning* dapat dilihat pada Tabel 2.

TABEL 2

HASIL PENGUJIAN MACHINE LEARNING

Machine Learning Algorithms	Accuracy	Error Rate	Recall	Specificity	F-1 Score	Precision
K-Nearest Neighbor	92.3%	7%	92.3%	92.3%	92.3%	92.3%
Random Forest	90.4%	9.6%	92.3%	88.5%	88.9%	88.9%
Support Vector Machine	73.1%	26.9%	84.6%	61.5%	75.9%	68.8%

Dibuktikan pada Tabel 2, didapatkan model dengan performansi terbaik pada penelitian ini yaitu model KNN. Dari 52 data uji, model KNN berhasil mendapatkan akurasi sebesar 92.3%, *error rate* sebesar 7%, *recall* sebesar 92.3%, spesifisitas sebesar 92.3%, dan *f-1 score* sebesar 92.3%. Hal ini menunjukkan bahwa model yang menggunakan algoritma KNN dapat memprediksi kanker ovarium dengan baik. Pengujian model tersebut dapat dilakukan dengan efektif, karena menggunakan *K generator* yang dapat menangani kekurangan dari algoritma KNN, yaitu tidak diketahui mana nilai *K* yang terbaik dan harus dilakukan uji coba nilai *K* satu-persatu.

B. Pengujian Akurasi dan Analisis pada Aplikasi

Pada pengujian ini akan dibahas tentang pengujian akurasi dan analisis pada aplikasi deteksi dini kanker ovarium untuk mengetahui hasil performansi sistem. Sistem deteksi kanker yang telah dibuat menggunakan metode KNN, diimplementasikan pada aplikasi android. Pengujian menggunakan sebanyak 172 data uji berupa hasil kuesioner gejala penyakit kanker ovarium yang didapatkan dari Rumah Sakit Hasan Sadikin. Untuk mengetahui akurasi pada aplikasi deteksi dini kanker ovarium digunakan Rumus (8).

$$Akurasi(\%) = \frac{138 \text{ data benar}}{172 \text{ data uji}} \times 100\%$$

$$Akurasi(\%) = 80,23\%$$

Dari pengujian yang telah dilakukan, didapatkan hasil akurasi pada aplikasi sebesar 80.23% dengan 138 data benar dan 34 data salah dari total 172 data uji. Berdasarkan hasil akurasi sistem, dapat dilihat bahwa performansi sistem pada aplikasi deteksi dini kanker ovarium telah cukup baik.

C. Pengujian dan Analisis *Quality of Service*

Model yang telah dibuat dapat langsung *deploy* pada *cloud server Digital Ocean*, lalu dilakukan pengujian QoS

pada sistem tersebut menggunakan tiga parameter, yaitu *latency*, *throughput*, dan *load performance*.

a. Pengujian dan Analisis *Latency*

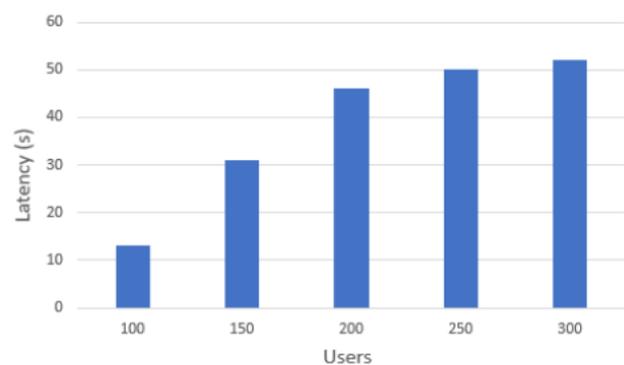
Pengujian *latency* pada Tugas Akhir ini diberikan beban *user* sebanyak 100, 150, 200, 250, dan 300 *user*, *ram-up* periode sebesar 20 *second*, dan *loop count* sebesar 1 kali. Hasil pengujian *latency* dapat dilihat pada Tabel 3.

TABEL 3

HASIL PENGUJIAN LATENCY

Pengujian	Jumlah User	Latency
Pengujian 1	100	13 second
Pengujian 2	150	31 second
Pengujian 3	200	46 second
Pengujian 4	250	50 second
Pengujian 5	300	52 second

Pada Tabel 3, didapatkan *latency* terbaik berada pada Pengujian 1 dengan jumlah *user* sebanyak 100 dengan hasil *latency* sebesar 13 *second*. Hasil pengujian juga dapat dilihat pada Gambar 9.



GAMBAR 9

GRAFIK HASIL PENGUJIAN LATENCY

Pada Grafik 5, dapat dilihat bahwa semakin banyak *user* yang melakukan permintaan, maka nilai *latency* semakin besar. Hal ini terjadi karena *server* memerlukan waktu lebih untuk menangani permintaan dari banyak user, selain itu antrian akses pada server menjadi semakin panjang. Dapat diketahui juga dari Gambar 9 bahwa rata-rata kenaikan *latency* per 50 *users* yaitu sebesar 12 *second*.

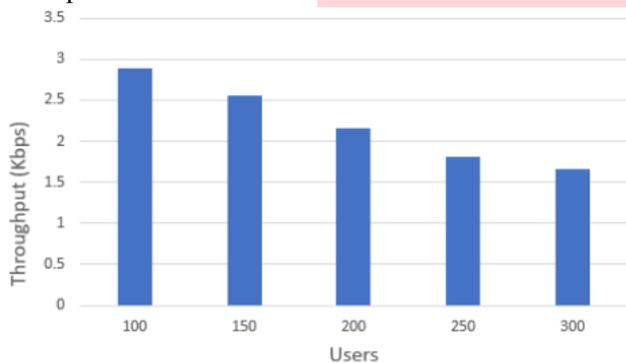
b. Pengujian dan Analisis *Throughput*

Pengujian *throughput* pada tugas akhir ini diberikan beban *user* sebanyak 100, 150, 200, 250, dan 300 *user*, *ram-up* periode sebesar 20 *second*, dan *loop count* sebesar 1 kali. Hasil pengujian *throughput* dapat dilihat pada Tabel 4.

TABEL 4
HASIL PENGUJIAN *THROUGHPUT*

Pengujian	Jumlah User	<i>Throughput</i>
Pengujian 1	100	2.89 kbps
Pengujian 2	150	2.56 kbps
Pengujian 3	200	2.16 kbps
Pengujian 4	250	1.81 kbps
Pengujian 5	300	1.66 kbps

Pada Tabel 4 didapatkan *throughput* terbaik berada pada Pengujian 1 dengan jumlah *user* sebanyak 100 dengan hasil *throughput* sebesar 2.89 kbps. Hasil pengujian juga dapat dilihat pada Gambar 10.



GAMBAR 10
GRAFIK HASIL PENGUJIAN *THROUGHPUT*

Dapat dilihat pada Gambar 10, bahwa nilai *throughput* terbaik berada pada Pengujian 1 dengan jumlah permintaan sebanyak 100 *user*. Lalu ketika semakin banyak *user* yang melakukan permintaan, nilai dari *throughput* menjadi semakin kecil. Hal ini dapat terjadi karena *server* yang digunakan tidak kuat dalam menangani permintaan dari banyak *user*. Dapat diketahui juga bahwa rata-rata penurunan *throughput* per 50 *users* yaitu sebesar 1.24 Kbps

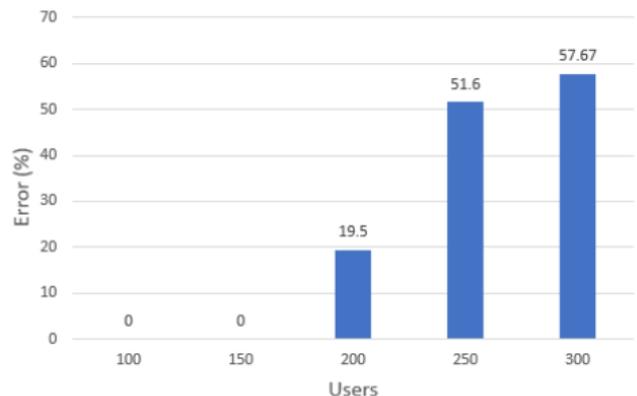
c. Pengujian dan Analisis *Load Performance*

Pengujian *load performance* pada tugas akhir ini diberikan beban *user* sebanyak 100, 150, 200, 250, dan 300 *user*, *ram-up* periode sebesar 20 *second*, dan *loop count* sebesar 1 kali. Hasil pengujian *load performance* dapat dilihat pada Tabel 5.

TABEL 5
HASIL PENGUJIAN *LOAD PERFORMANCE*

Pengujian	Jumlah User	<i>Error</i>
Pengujian 1	100	0.00%
Pengujian 2	150	0.00%
Pengujian 3	200	19.50%
Pengujian 4	250	51.60%
Pengujian 5	300	57.67%

Pada Tabel 5, didapatkan *load performance* terbaik berada pada Pengujian 1 dan Pengujian 2 dengan hasil *error* masing-masing sebesar 0.00%. Hasil pengujian juga dapat dilihat pada Gambar 11.



GAMBAR 11
GRAFIK HASIL PENGUJIAN *LOAD PERFORMANCE*

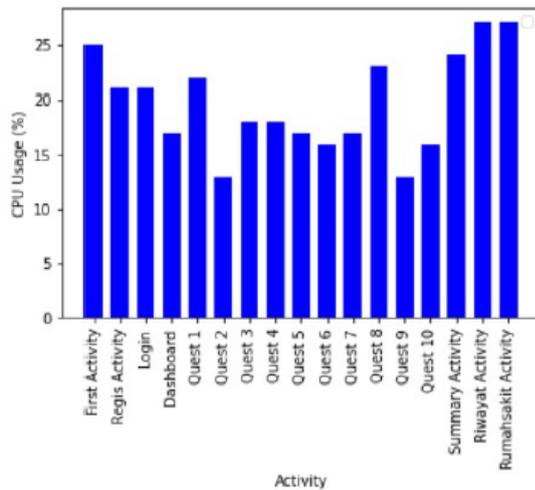
Pada Gambar 11, dapat dilihat pada jumlah permintaan dari 100 dan 150 *user* tidak mengalami *error*, sedangkan pada jumlah permintaan 200, 250, dan 300 *user* jumlah *error* semakin meningkat. Dapat disimpulkan bahwa *server* yang digunakan mampu menangani jumlah permintaan masuk dari 100 dan 150 *user* dengan baik karena tidak terdapat *error*, sedangkan pada jumlah permintaan 200, 250, dan 300 *server* sudah tidak mampu menangani semua permintaan dengan baik. Dapat diketahui bahwa rata-rata *error* per 50 *users* yaitu sebesar 57.67%.

D. Pengujian CPU Usage dan Memory Usage

Pengujian CPU Usage dan Memory Usage dilakukan menggunakan aplikasi melalui smartphone android dengan mengoperasikan semua fitur yang terdapat pada aplikasi. Pengujian ini bertujuan untuk mengetahui seberapa banyak aplikasi mengonsumsi CPU dan *memory*. Perangkat yang digunakan dalam pengujian ini adalah Samsung Galaxy J3 Pro dengan spesifikasi Android 9.0, 16GB penyimpanan, dan 2GB RAM (*memory*).

a. Pengujian dan Analisis CPU Usage

Pengujian CPU Usage dilakukan pada masing-masing activity pada aplikasi android. Terdapat 17 *activity* pada aplikasi yang dibuat yaitu *First Activity*, *Regis Activity*, *Login*, *Dashboard*, *Quest1*, *Quest2*, *Quest3*, *Quest4*, *Quest5*, *Quest6*, *Quest7*, *Quest8*, *Quest9*, *Quest10*, *Summary Activity*, *Riwayat Activity*, dan *Rumahsakit Activity*. Hasil pengujian CPU usage dapat dilihat pada Gambar 12.

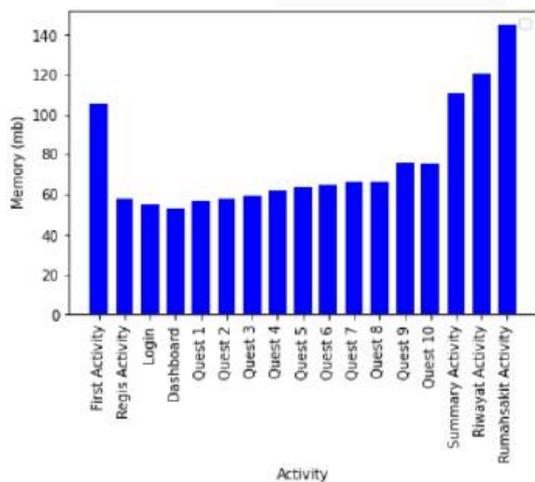


GAMBAR 12
GRAFIK HASIL PENGUJIAN CPU USAGE

Dari Gambar 8, dapat dilihat bahwa *activity* yang paling banyak penggunaan CPU pada perangkat android adalah Riwayat *Activity* dan Rumahsakit *Activity*. Dari hasil pengujian tersebut dapat diketahui bahwa rata-rata penggunaan CPU pada perangkat android adalah 19.7%. Dari hasil tersebut dapat dikatakan bahwa aplikasi deteksi dini kanker ovarium dapat berjalan dengan baik pada perangkat *smartphone* tanpa adanya *lag* dengan hanya mengonsumsi 19.7% CPU.

b. Pengujian dan Analisis *Memory Usage*

Pada pengujian *memory usage* juga dilakukan pengujian pada masing-masing *activity* pada aplikasi android. Hasil pengujian *memory usage* dapat dilihat pada Gambar 13.



GAMBAR 13
GRAFIK HASIL PENGUJIAN MEMORY USAGE

Dari Gambar 13, dapat dilihat bahwa *activity* yang paling banyak penggunaan *memory* pada perangkat android adalah Rumah Sakit *Activity*. Dari hasil pengujian tersebut dapat diketahui bahwa rata-rata penggunaan *memory* pada perangkat android adalah 76 MB. Dari hasil tersebut dapat dikatakan bahwa aplikasi deteksi dini kanker ovarium dapat

berjalan dengan baik pada perangkat *smartphone* dengan hanya mengonsumsi 76MB dari total *memory* 2GB.

E. Pengujian *Black Box*

Pengujian *black box* atau *black box testing* adalah pengujian pada perangkat lunak (*software*) yang berfokus hanya pada fungsionalitas *software* tersebut tanpa mengetahui proses yang terjadi di belakangnya. Pengujian *black box* dilakukan di akhir pembuatan *software* untuk mengetahui apakah *software* tersebut dapat berfungsi dengan baik. Hasil dari pengujian *black box* dapat dilihat pada Tabel 6, 7, 8, 9, 10, dan 11.

TABEL 6
PENGUJIAN HALAMAN LOGIN

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
1	Email dan Password tidak diisi kemudian klik tombol masuk	-Email : Kosong -Password : Kosong	Tidak bisa masuk dan akan memberikan pesan "Email tidak boleh kosong" "Password tidak boleh kosong".	Sesuai Harapan
2	Input Email dan Password yang tidak sesuai, kemudian klik tombol masuk	-Email : zu@em.co -Pass : Ijdwk02	Tidak bisa login dan akan memberikan pesan "There is no user record corresponding to this identifier".	Sesuai Harapan
3	Input Email dan Password yang sesuai, kemudian klik tombol masuk	-Email : zuhairi@eova.com -Pass : 123456	Sistem menerima akses login dan akan memberikan pesan "Selamat datang zuhairi@eova.com" kemudian menampilkan halaman utama.	Sesuai Harapan
4	Menampilkan Halaman Registrasi.	Klik teks "Daftar di Sini"	Akan menampilkan halaman registrasi.	Sesuai Harapan

TABEL 7
PENGUJIAN HALAMAN REGISTRASI

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
1	Masuk ke halaman registrasi.	Klik teks "Daftar di Sini" pada halaman login.	Akan menampilkan halaman registrasi.	Sesuai Harapan
2	Tidak mengisi salah satu atau seluruh data.	Mengosongkan salah satu atau seluruh data yang diminta.	Sistem akan menolak dan memberikan pesan "Email/Password tidak boleh kosong"	Sesuai Harapan
3	Input Email yang telah terdaftar	Input email yang telah didaftarkan sebelumnya.	Sistem menolak dan memberikan pesan "The email address is already use by another account"	Sesuai Harapan
4	Input Email dengan format yang tidak sesuai.	Email : ggh Pass : Kosong	Sistem menolak dan memberikan pesan "Email tidak sesuai"	Sesuai Harapan
5	Input seluruh data dengan benar.	Input seluruh data sesuai dengan yang diminta secara benar	Sistem menerima registrasi dan akan diarahkan ke halaman login.	Sesuai Harapan

TABEL 8
PENGUJIAN HALAMAN UTAMA

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
1	Menampilkan halaman test kanker ovarium.	Klik tombol "Mulai Test" pada halaman utama.	Akan menampilkan halaman test kanker ovarium.	Sesuai Harapan
2	Menampilkan halaman riwayat.	Klik tombol "Lihat riwayat" pada halaman utama.	Akan menampilkan halaman riwayat.	Sesuai Harapan
3	Menampilkan halaman daftar rumah sakit	Klik tombol "Lihat daftar rumah sakit" pada halaman utama.	Akan menampilkan halaman daftar rumah sakit	Sesuai Harapan
4	Keluar dari Aplikasi (Logout)	Klik tombol "Logout" pada halaman utama.	Akan keluar dari aplikasi dan menampilkan halaman login.	Sesuai Harapan

TABEL 9
PENGUJIAN HALAMAN TES KANKER OVARIUM

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
1	Menampilkan halaman pertanyaan dari satu sampai sepuluh	Klik tombol "Ya" atau "Tidak" pada halaman tes kanker ovarium.	Akan menampilkan sepuluh halaman pertanyaan.	Sesuai Harapan
2	Menampilkan halaman hasil diagnosa.	Klik tombol "Ya" atau "Tidak" pada halaman tes kanker ovarium sebanyak 10 kali.	Akan menampilkan halaman hasil diagnosa	Sesuai Harapan
3	Menampilkan halaman daftar rumah sakit.	Klik tombol "Cek rumah sakit terdekat" pada halaman hasil diagnosa.	Akan menampilkan halaman daftar rumah sakit	Sesuai Harapan
4	Menampilkan halaman utama.	Klik tombol "Halaman Utama" pada halaman hasil diagnosa.	Akan menampilkan halaman utama.	Sesuai Harapan

TABEL 10
PENGUJIAN HALAMAN RIWAYAT

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
1	Menampilkan halaman riwayat.	Klik tombol "Lihat Riwayat" pada halaman tes utama.	Akan menampilkan halaman riwayat.	Sesuai Harapan

TABEL 11
PENGUJIAN HALAMAN CEK DAFTAR RUMAH SAKIT

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
1	Menampilkan halaman Daftar Provinsi.	Klik tombol "Lihat Daftar Rumah Sakit" pada halaman utama.	Akan menampilkan halaman Daftar Provinsi.	Sesuai Harapan
2	Menampilkan halaman Daftar Rumah Sakit di Provinsi yang dipilih.	Klik provinsi yang ingin dipilih	Akan menampilkan halaman Daftar Rumah Sakit pada Provinsi.	Sesuai Harapan
3	Menampilkan Maps menuju rumah sakit yang dipilih.	Klik rumah sakit yang ingin dipilih.	Akan menampilkan halaman Google Maps menuju rumah sakit yang dipilih	Sesuai Harapan

Dari tabel-tabel tersebut dapat dilihat bahwa aplikasi deteksi dini kanker ovarium dapat berfungsi dengan baik tanpa adanya *error* di setiap *activity*nya.

V. KESIMPULAN

Perancangan sebuah sistem untuk mendeteksi kanker ovarium dengan sistem pakar berhasil dilakukan. Perancangan sistem dapat dibuat menggunakan tiga algoritma *machine learning* yaitu KNN, SVM, dan *Random Forest*. Sistem yang dirancang menggunakan algoritma KNN, dapat mendeteksi resiko kanker ovarium dengan baik. Hal ini dapat dibuktikan dari hasil pengujian performansi sistem, dengan hasil akurasi sebesar 92%, *error rate* sebesar 7%, *recall* sebesar 92.3% dan spesifisitas sebesar 92.3%.

Sistem deteksi kanker menggunakan KNN tersebut berhasil diimplementasikan pada aplikasi Android, lalu dilakukan pengujian akurasi menggunakan 172 data uji dan didapatkan hasil akurasi sebesar 80.23%. Hal ini menunjukkan bahwa sistem yang telah diimplementasikan pada aplikasi Android mampu mendeteksi kanker ovarium dengan cukup baik.

Aplikasi deteksi dini kanker ovarium dapat berjalan dengan baik dan lancar pada *smartphone*, dibuktikan dengan pengujian CPU dan *memory usage*, aplikasi deteksi dini kanker ovarium hanya menghabiskan 19.7% CPU dan memakan sebanyak 76MB dari total 2GB *memory*. Lalu dibuktikan dengan pengujian *black box*, aplikasi deteksi dini kanker ovarium dapat berfungsi sesuai dengan fungsinya tanpa adanya *error*.

Cloud Server yang digunakan pada penelitian ini bisa dikatakan cukup baik karena mampu menangani permintaan dari 100 sampai 150 users tanpa *error*. Namun, ketika melebihi 100 dan 150 users, server belum mampu menangani permintaan secara optimal. Terbukti dari pengujian QoS, *error*

semakin meningkat ketika jumlah permintaan dari user semakin banyak.

VI. REFERENSI

- [1] W. P. Prakosa, "Laporan studi kasus asuhan keperawatan pada kanker ovarium dengan kecemasan," Ph.D. dissertation, STIKES Muhammadiyah Klaten, 2021.
- [2] D. U. Rosyidah, "Peningkatan pengetahuan tentang kanker leher rahim dan deteksi dini dengan pemeriksaan pap smear," *Jurnal Pengabdian Masyarakat Medika*, 2021.
- [3] L. K. Sari, H. Fauzi, and K. N. Ramadhani, "Perancangan sistem aplikasi self-awareness kanker ovarium berbasis android menggunakan metode sistem pakar," *eProceedings of Engineering*, vol. 8, no. 2, 2021.
- [4] A. B. Harsono, "Kanker ovarium:the silent killer," *Indonesian Journal of Obstetrics & Gynecology Science*, vol. 3, no. 1, pp. 1–6, 2020.
- [5] M. Dahria, "Pengembangan sistem pakar dalam membangun suatu aplikasi." 2011, 199-205.
- [6] P. D. Kusuma. Deepublish Publisher, 2020, vol. 1.
- [7] A. U. Zailani, A. Perdananto, N. Nurjaya, and S. Sholihin, "Pengenalan sejak dini siswa smp tentang machine learning untuk klasifikasi gambar dalam menghadapi revolusi 4.0," *KOMMAS: Jurnal Pengabdian Kepada Masyarakat*, vol. 1, no. 1, 2020.
- [8] J. J. Pangaribuan and V. Angkasa, "Komparasi tingkat akurasi random forest dan knn untuk mendiagnosis penyakit kanker payudara," *Journal Information System Development (ISD)*, vol. 7, no. 1, pp. 49–61, 2022.
- [9] R. C. Putra, "Pembangunan perangkat pendeteksi jenis gerakan raket bulu tangkis dengan algoritma knn dan svm," *Jurnal Teknik*, vol. 9, no. 2, 2020.
- [10] U. Hasanah, L. R. Mayangsari, A. Pratama, and I. Cholissodin, "Perbandingan metode svm, fuzzy-knn, dan bdt-svm untuk klasifikasi detak jantung hasil elektrokardiografi," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 3, no. 3, pp. 201–207, 2016.
- [11] R. A. Haristu, "Penerapan metode random forest untuk prediksi win ratio pada pemain player unknown battleground," *Universitas Sanata Dharma Yogyakarta*, vol. 1, no. 2, 2019.
- [12] R. K. Ngantung and M. I. Pakereng, "Model pengembangan sistem informasi akademik berbasis user centered design menerapkan framework flask python," *Jurnal Media Informatika Budidarma*, vol. 5, no. 3, pp. 1052–1062, 2021.
- [13] R. Prasthio, Y. Yohannes, and S. Devella, "Penggunaan fitur hog dan hsv untuk klasifikasi citra sel darah putih," *Jurnal Algoritme*, vol. 2, no. 2, pp. 120–132, 2022.
- [14] M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: an overview," *arXiv preprint arXiv:2008.05756*, 2020.