

MEMBUAT APLIKASI PREDIKSI KANKER PAYUDARA TANPA PUSING

Menggunakan Support Vector Machine
Dengan Framework Django
Amatir Auto Mahir



**CAHYO PRIANTO
RESA RANTI
NUR TRI RAMADHANTI A**

MEMBUAT APLIKASI PREDIKSI KANKER PAYUDARA TANPA PUSING

MEMBUAT APLIKASI PREDIKSI KANKER PAYUDARA TANPA PUSING

Cahyo Prianto
Resa Rianti
Nur Tri Ramadhanti Adiningrum



MEMBUAT APLIKASI PREDIKSI KANKER PAYUDARA TANPA PUSING

© BUKU PEDIA

Penulis:
Cahyo Prianto
Resa Rianti
Nur Tri Ramadhanti Adiningrum

Editor:
(Nama Penguji Sidang)

Cetakan Pertama: Isi dengan Bulan saat upload buku

Cover: Tim Penyusun

Tata Letak: Tim Kreatif Penerbit

Hak Cipta 2023, pada Penulis. Diterbitkan pertama kali oleh:

BUKU PEDIA
Jl. Sariasih No. 54
Bandung 40151

Website: BUKU PEDIA
E-mail: penerbit@bukupedia.co.id

Copyright © 2023 by BUKU PEDIA
All Right Reserved

- Cet. I -: BUKU PEDIA, TAHUN TERBIT
Dimensi : 14,8 x 21 cm
ISBN: KOSONGKAN DULU

Hak cipta dilindungi undang-undang
Dilarang memperbanyak buku ini dalam bentuk dan dengan cara apapun tanpa izin tertulis dari
penulis dan penerbit

Undang-undang No.19 Tahun 2002 Tentang
Hak Cipta Pasal 72

Undang-undang No.19 Tahun 2002 Tentang Hak Cipta
Pasal 72

Barang siapa dengan sengaja melanggar dan tanpa hak melakukan perbuatan sebagaimana dimaksud dalam pasal ayat (1) atau pasal 49 ayat (1) dan ayat (2) dipidana dengan pidana penjara masing-masing paling sedikit 1 (satu) bulan dan/atau denda paling sedikit Rp.1.000.000,00 (satu juta rupiah), atau pidana penjara paling lama 7 (tujuh) tahun dan/atau denda paling banyak Rp.5.000.000.000,00 (lima miliar rupiah).

Barang siapa dengan sengaja menyiarakan, memamerkan, mengedarkan, atau menjual kepada umum suatu ciptaan atau barang hasil pelanggaran hak cipta terkait sebagai dimaksud pada ayat (1) dipidana dengan pidana penjara paling lama 5 (lima) tahun dan/atau denda paling banyak Rp.500.000.000,00 (lima ratus juta rupiah).

KATA PENGANTAR

Segala puji bagi Allah, Tuhan Yang Maha Esa atas rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan buku ini. Tak lupa juga mengucapkan salawat serta salam semoga senantiasa tercurahkan kepada Nabi Besar Muhammad SAW, karena berkat beliau, kita mampu keluar dari kegelapan menuju jalan yang lebih terang.

Kami ucapan juga rasa terima kasih kami kepada pihak-pihak yang mendukung lancarnya buku ini mulai dari proses penulisan hingga proses cetak, yaitu orang tua kami, rekan-rekan kami, penerbit, dan masih banyak lagi yang tidak bisa kami sebutkan satu per satu.

Adapun, buku kami yang berjudul Membuat Aplikasi Prediksi Kanker Payudara Tanpa Pusing ini telah selesai kami buat secara semaksimal dan sebaik mungkin agar menjadi manfaat bagi pembaca yang membutuhkan informasi dan pengetahuan mengenai bagaimana memprediksi kanker payudara. Dalam buku ini, tertulis bagaimana cara olah data serta pembuatan aplikasi yang menjadi alternatif pegangan bagi mahasiswa dan pihak yang memerlukan.

Kami sadar, masih banyak luput dan kekeliruan yang tentu saja jauh dari sempurna tentang buku ini. Oleh sebab itu, kami mohon agar pembaca memberi kritik dan juga saran terhadap karya buku ajar ini agar kami dapat terus meningkatkan kualitas buku. Demikian buku ajar ini kami buat, dengan harapan agar pembaca dapat memahami informasi dan juga mendapatkan wawasan mengenai bidang sistem informasi manajemen serta dapat bermanfaat bagi masyarakat dalam arti luas. Terima kasih.

Bandung, 02 November 2022
Penulis

DAFTAR ISI

KATA PENGANTAR.....	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xiii
BAB 1 Apa Itu Kanker Payudara?.....	1
1.1 Apa Itu Kanker?.....	1
1.2 Apa Itu Payudara?.....	2
1.3 Apa itu Kanker Payudara?.....	3
1.4 Penyebab Kanker Payudara.....	4
1.5 Gejala Kanker Payudara	4
1.6 Cara Pengobatan Kanker Payudara	5
1.7 Cara Mencegah Kanker Payudara.....	6
1.8 Kasus Yang Pernah Terjadi.....	6
BAB 2 Metodologi Pembuatan Aplikasi.....	8
2.1 Metodologi Pengolahan Data	8
2.2 <i>Business Understanding</i>	9
2.3 <i>Data Understanding</i>	9
2.3.1 Tipe Data	10
2.3.2 Kualitas pada dataset	10
2.3.3 Dataset Yang Digunakan	13
2.4 <i>Data Preparation</i>	21
2.5 <i>Modelling</i>	26
2.5.1 Support Vector Machine.....	27
2.6 <i>Evaluation</i>	32

2.7 <i>Deployment</i>	34
BAB 3 Apa Saja Perangkat Yang Digunakan Untuk Membuat Aplikasi?	35
3.1 Bahasa Pemrograman dan Pendukung Yang Digunakan	35
3.1.1 Python	35
3.1.2 HTML.....	37
3.1.3 CSS	38
3.1.4 Bootstrap	39
3.1.5 Django	40
3.2 Penyimpanan Yang Digunakan	41
3.2.1 Database MySQL	41
3.3 Perangkat Lunak Yang Dibutuhkan	43
3.3.1 Visual Studio Code	43
3.3.2 XAMPP	45
3.3.3 Jupyter Notebook.....	47
3.3.4 Tableau.....	48
3.3.5 Anaconda	50
3.3.6 Kaggle	51
3.4 Perangkat Keras Yang Dibutuhkan.....	52
BAB 4 Perancangan Sistem UML dalam Membuat Aplikasi.....	53
4.1 Analisis.....	53
4.1.1 Analisis Sistem Yang Akan di Bangun	54
4.2 Perancangan Sistem (UML).....	56
4.2.1 Use case Diagram	56
4.2.2 Class Diagram	64
4.3 Struktur Menu.....	85
BAB 5 Implementasi Aplikasi	87
5.1 Implementasi Kode Aplikasi.....	87

5.1.1	Implementasi Package Yang Digunakan.....	87
5.1.2	Implementasi Proses Login.....	89
5.1.3	Implementasi Proses Registrasi.....	90
5.1.4	Implementasi Proses Prediksi Kanker Payudara..	91
5.1.5	Implementasi Hasil Prediksi Kanker Payudara ..	93
5.1.6	Implementasi Proses CRUD Data Kanker Payudara	
	93	
5.1.7	Implementasi Proses Menampilkan Visualisasi Data	
	101	
5.1.8	Implementasi Proses Logout.....	102
5.2	Implementasi Halaman Antar Muka	102
5.2.1.	Implementasi Halaman Depan	102
5.2.2.	Implementasi Halaman Login.....	103
5.2.3.	Implementasi Halaman Registrasi.....	104
5.2.4.	Implementasi Halaman Prediksi Kanker Payudara	
	105	
5.2.5.	Implementasi Halaman Hasil Prediksi Kanker	
	Payudara	107
5.2.6.	Implementasi Halaman Menampilkan Data Kanker	
	Payudara	107
5.2.7.	Implementasi Halaman Tambah Data Kanker	
	Payudara	109
5.2.8.	Implementasi Halaman Edit Data Kanker Payudara	
	110	
5.2.9.	Implementasi Halaman Visualisasi.....	111
BAB 6	Kesimpulan dan Saran.....	112
6.1	Kesimpulan	112
6.2	Saran	113
DAFTAR PUSTAKA.....		114

BIODATA PENULIS..... 117

DAFTAR GAMBAR

Gambar 1. 1 Sel Kanker	1
Gambar 1. 2 Ilustrasi Payudara	2
Gambar 1. 3 Ilustrasi Lambang Kanker Payudara	3
Gambar 2. 1 Metodologi CRISP-DM	8
Gambar 2. 2 Struktur Dataset.....	10
Gambar 2. 3 Situs Download Data Kanker Payudara	13
Gambar 2. 4 Visualisasi SVM.....	28
Gambar 2. 5 Situs Download Python.....	36
Gambar 3. 1 Logo Python	35
Gambar 3. 2 Logo HTML.....	37
Gambar 3. 3 Logo CSS	38
Gambar 3. 4 Logo Bootstrap	39
Gambar 3. 5 Situs Donwload Bootstrap	39
Gambar 3. 6 Logo Django	40
Gambar 3. 7 Situs Donwload Django	40
Gambar 3. 8 Logo MySQL	41
Gambar 3. 9 Logo Visual Studio Code	43
Gambar 3. 10 Situs Download Visual Studio Code.....	43
Gambar 3. 11 Logo XAMPP	45
Gambar 3. 12 Situs Download XAMPP	45
Gambar 3. 13 Logo Jupyter Notebook.....	47
Gambar 3. 14 Situs Download Jupyter Notebook.....	47
Gambar 3. 15 Logo Tableau.....	48
Gambar 3. 16 Situs Download Tableau.....	49
Gambar 3. 17 Logo Anaconda	50
Gambar 3. 18 Situs Download Anaconda.....	50
Gambar 3. 19 Logo Kaggle	51
Gambar 3. 20 Situs Resmi Kaggle	51
Gambar 4. 1 Flowmap Login Aplikasi	54
Gambar 4. 2 Flowmap Prediksi Kanker Payudara	55
Gambar 4. 3 Use Case Diagram	57
Gambar 4. 4 Class Diagram	66
Gambar 4. 5 Activity diagram Login Admin.....	68
Gambar 4. 6 Activity diagram Melakukan Registrasi Admin.....	69
Gambar 4. 7 Activity diagram Melakukan Pengolahan Data	71
Gambar 4. 8 Activity diagram Melakukan Prediksi Kanker Payuda	

.....	72
Gambar 4. 9 Activity diagram Melakukan Visualisasi	73
Gambar 4. 10 Activity diagram Logout Admin.....	75
Gambar 4. 11 Statechart Diagram melakukan login admin.....	76
Gambar 4. 12 Statechart Diagram melakukan registrasi admin .	77
Gambar 4. 13 Statechart Diagram melakukan prediksi kanker payudara.....	78
Gambar 4. 14 Statechart Diagram melakukan pengolahan data	80
Gambar 4. 15 Statechart Diagram melakukan visualisasi data..	82
Gambar 4. 16 Component Diagram.....	83
Gambar 4. 17 Entity Relationship Diagram aplikasi prediksi kanker payudara	85
Gambar 4. 18 Struktur Menu	85
Gambar 5. 1 Halaman Depan Aplikasi	102
Gambar 5. 2 Halaman Login Aplikasi	103
Gambar 5. 3 Halaman Registrasi Aplikasi	104
Gambar 5. 5 Halaman Prediksi Kanker	105
Gambar 5. 6 Halaman Hasil Prediksi Aplikasi	107
Gambar 5. 7 Halaman Data Kanker Payudara	107
Gambar 5. 8 Halaman Tambah Data Kanker Payudara Aplikasi	109
Gambar 5. 9 Halaman Edit Data Kanker Payudara	110
Gambar 5. 10 Halaman Visualisasi Aplikasi	111

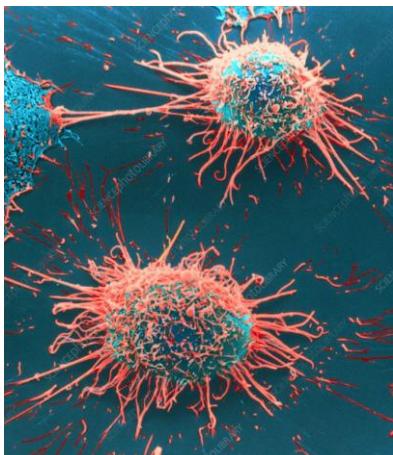
DAFTAR TABEL

Tabel 2. 1 Atribut dan Artinya dari Dataset	14
Tabel 3. 1 Spesifikasi dan Kebutuhan Perangkat Keras	52
Tabel 4. 1 Definisi Aktor	58
Tabel 4. 2 Definisi Use Case	58
Tabel 4. 3 Skenario Use case Machine Learning	60
Tabel 4. 4 Skenario Use case Login Admin	60
Tabel 4. 5 Skenario Use case Registrasi.....	61
Tabel 4. 6 Skenario Use case Melakukan Prediksi.....	62
Tabel 4. 7 Skenario Use case Melakukan Visualisasi Data	62
Tabel 4. 8 Skenario Use case Melakukan Pengolahan Data	63
Tabel 4. 9 Skenario Use case Logout Admin.....	64

BAB 1

Apa Itu Kanker Payudara?

1.1 Apa Itu Kanker?



Gambar 1. 1 Sel Kanker
Sumber:<https://www.sciencephoto.com/media/294814/view/cervical-cancer-cells-sem>

Kanker adalah sel yang tumbuh secara tidak terkendali dan tidak normal (abnormal) secara terus-menerus dan tidak terbatas. Pertumbuhan sel-sel kanker tidak terkoordinasi dengan jaringan lain dan dapat merusak tubuh. Kanker juga dikenal sebagai tumor ganas yang mengalami pertumbuhan abnormal yang tidak diketahui secara pasti penyebabnya. Sel normal hanya akan terus berkembang melalui proses pembelahan diri

saat terjadi kematian atau kerusakan. Namun, sel kanker akan terus tumbuh meskipun tidak dibutuhkan oleh tubuh dan dapat merusak jaringan sel normal lainnya serta menyebar ke organ tubuh lain melalui jaringan ikat, darah, saraf, dan jaringan penunjang organ tubuh. Kanker adalah penyakit yang tidak menular yang timbul akibat kondisi fisik yang tidak normal dan pola hidup yang tidak sehat[1].

1.2 Apa Itu Payudara?



Gambar 1. 2 Ilustrasi Payudara

Sumber :<https://www.dandad.org/awards/professional/2018/press-advertising-campaigns-local/>

Payudara, juga dikenal dengan nama mammae, adalah organ tubuh bagian atas dada pada spesies mamalia betina, termasuk perempuan manusia. Biasanya, payudara digunakan untuk menyalurkan air susu (ASI) pada bayi mamalia atau manusia yang baru lahir. Payudara memiliki nilai estetika karena bentuknya yang menonjol di bagian atas tubuh, membuat tubuh wanita terlihat lebih menarik. Fungsi utama payudara adalah sebagai produsen ASI dari lobulus-lobulus, yaitu kelenjar yang menghasilkan ASI, dan tubulus atau ductus yang mengantarkan ASI dari kelenjar ke putting susu (nipple). Kelenjar mammae merupakan ciri khas dari semua mamalia. Bentuknya mirip kerucut, namun tidak semua wanita memiliki ukuran yang sama. Jadi, payudara merupakan organ vital yang memproduksi ASI dalam jumlah besar[2].

1.3 Apa itu Kanker Payudara?

Kanker payudara adalah kanker yang paling banyak ditemukan pada wanita dan angka kematiannya masih menempati posisi kedua diantara kanker lainnya[3]. Carsinoma Mammae, lebih dikenal sebagai kanker payudara, adalah penyakit di mana sel-sel kanker terbentuk di jaringan payudara. Kanker dapat muncul di kelenjar yang menghasilkan susu (lobulus), saluran yang membawa susu dari kelenjar ke puting (duktus), atau di jaringan lemak atau ikat di dalam payudara. Kanker payudara adalah penyebab kematian kedua pada wanita dan dapat menyebabkan gejala psikologis seperti kecemasan, depresi, panik, dan gegar otak pada pasiennya. Penyebab pasti dari kanker payudara masih belum diketahui, namun dianggap multifaktorial. Upaya pencegahan seperti pendidikan masyarakat dan remediasi diperlukan untuk menangani masalah ini, sesuai dengan kebutuhan individu. Mencegah kanker payudara sejak dini melalui SADARI (Pemeriksaan Payudara Sendiri) adalah hal yang penting yang harus dilakukan oleh setiap orang. Namun, jika kanker payudara terdiagnosis, pemeriksaan lebih lanjut oleh dokter diperlukan[4].



Gambar 1. 3 Ilustrasi Lambang Kanker Payudara
Sumber :

https://www.freepik.com/premium-vector/cancer-design-illustration_3

1.4 Penyebab Kanker Payudara

Faktor penyebab yang memungkinkan untuk timbulnya kanker payudara antara lain :

- a. Mengkonsumsi makanan yang mengandung lemak berlebih dan berprotein tinggi, tetapi serat terlalu banyak. Dengan mengkonsumsi makanan seperti itu mengandung zat karsinogen yang dapat merangsang pertumbuhan sel kanker.
- b. Pada usia muda telah mengkonsumsi pil kontrasepsi. Penelitian telah membuktikan jika usia dini (remaja) sudah meminum oral (pil) sangat tinggi risikonya terserang kanker payudara
- c. Keturunan dari anggota keluarga yang pernah terkena kanker payudara.
- d. Mengkonsumsi alkohol terlalu banyak.
- e. Wanita merokok memiliki risiko paling besar dapat terserang kanker payudara karena kontaminasi senyawa kimia yang terlalu berlebihan

1.5 Gejala Kanker Payudara

Pada stadium awal, penyakit ini tidak dapat menunjukkan gejala tertentu. Oleh karena itu, sangat penting untuk melakukan pemeriksaan payudara sendiri setiap bulan, sepuluh hari setelah akhir periode perempuan. Rasakan rabaan searah jarum jam dengan hati-hati untuk mendeteksi adanya benjolan atau perubahan pada payudara[5].

Beberapa gejala kanker payudara yang bisa dialami oleh penderitanya adalah:

- a. Benjolan atau pengerasan payudara yang berbeda dari jaringan di sekitarnya.
- b. Darah keluar dari puting.
- c. Kemerahan atau penggunaan pori-pori kulit payudara yang menyerupai kulit jeruk.

- d. Nyeri dan bengkak pada payudara.
- e. Pengelupasan kulit di sekitar puting.
- f. Perubahan pada kulit payudara, seperti depresi.
- g. Perubahan ukuran, bentuk, atau tampilan payudara.
- h. Puting ditarik (ditarik atau dibalik) ke dalam.
- i. Benjolan atau pembengkakan di ketiak bagian bawah.

Ada banyak penyebab munculnya benjolan di payudara, yang sebagian besar bukan kanker. 90% dari massa payudara tidak bersifat kanker. Gangguan payudara non-kanker termasuk massa jinak seperti fibroadenoma dan kista serta infeksi[6].

1.6 Cara Pengobatan Kanker Payudara

Perawatan untuk penyakit ini tergantung pada jenis kanker yang Anda derita, stadium kanker, ukuran massa, dan sensitivitas sel kanker terhadap hormon. Pengobatan kanker payudara bisa sangat efektif, mencapai probabilitas kelangsungan hidup 90% atau lebih tinggi, terutama ketika penyakit ini diidentifikasi lebih awal[6]. Dokter akan menentukan terapi yang paling tepat untuk kondisi pasien berdasarkan hal-hal tersebut[5]. Ada beberapa pilihan pengobatan untuk kanker payudara, antara lain:

- a. Pembedahan, yang meliputi pengangkatan kanker atau benjolan (lumpektomi), pengangkatan seluruh payudara (mastektomi), pengangkatan sejumlah kelenjar getah bening (biopsi simpul sentinel), atau pengangkatan beberapa kelenjar getah bening (diseksi kelenjar getah bening aksila).
- b. Radioterapi, yang menggunakan sinar-X dan proton untuk membunuh sel kanker.
- c. Kemoterapi, yaitu dilakukan dengan menggunakan obat-obatan tertentu untuk membunuh sel kanker.
- d. Terapi hormonal, untuk mencegah sel kanker mendapatkan hormon yang mereka butuhkan untuk tumbuh.
- e. Terapi biologis, yang bekerja dengan sistem kekebalan pasien untuk membantu melawan sel kanker, atau untuk mengontrol

efek samping dari perawatan kanker lainnya.

- f. Terapi radiasi menggunakan sinar berenergi tinggi untuk membunuh sel kanker.

1.7 Cara Mencegah Kanker Payudara

Kanker payudara memang sangat mengerikan. Penyakit ini bisa disebabkan oleh banyak faktor seperti faktor keturunan, faktor lingkungan, dan gaya hidup yang tidak sehat. Untuk mencegah kanker payudara, sebaiknya periksa kondisi payudara secara rutin dengan cara menyentuh dan memijatnya. Cara ini cukup efektif untuk mendeteksi pertumbuhan sel kanker payudara sebelum mencapai stadium lanjut[7]. Kanker ini juga dapat dicegah dengan cara-cara yang dijelaskan di bawah ini.

- a. Berolahraga minimal 30 menit sehari
- b. Berhenti merokok
- c. Konsumsi buah dan sayur secara teratur
- d. Konsumsi produk kedelai
- e. Asupan serat makanan
- f. Hindari alkohol
- g. Kurangi asupan gula
- h. Hindari konsumsi lemak jahat
- i. Mengetahui riwayat orang tua
- j. Hindari memakan daging merah
- k. Mengkonsumsi obat-obatan
- l. Operasi

1.8 Kasus Yang Pernah Terjadi

Diperkirakan ada 18,1 juta kasus kanker di seluruh dunia pada tahun 2020. Dari jumlah tersebut, 9,3 juta kasus terjadi pada pria dan 8,8 juta pada wanita. Kanker yang paling umum secara global adalah kanker payudara dengan total 2.261.419 kasus atau sebanyak 12.5% dari total persentase berbagai kanker dan merupakan kanker yang menempati posisi pertama kasus kanker

terbayak di dunia[8].

Pada tahun 2020, terdapat 2,3 juta wanita yang didiagnosis menderita kanker payudara dan 685.000 kematian secara global. Pada akhir tahun 2020, terdapat 7,8 juta wanita hidup yang didiagnosis menderita kanker payudara dalam 5 tahun terakhir, kanker paling umum di dunia. Ada lebih banyak kehidupan yang berakhir akibat oleh wanita karena kanker payudara secara global daripada jenis kanker lainnya. Kanker payudara terjadi di setiap negara di dunia pada wanita pada usia berapa pun setelah pubertas tetapi dengan tingkat yang meningkat di kemudian hari[6].

Kelangsungan hidup kanker payudara selama minimal 5 tahun setelah diagnosis berkisar dari lebih dari 90% di negara-negara berpenghasilan tinggi, hingga 66% di India dan 40% di Afrika Selatan.

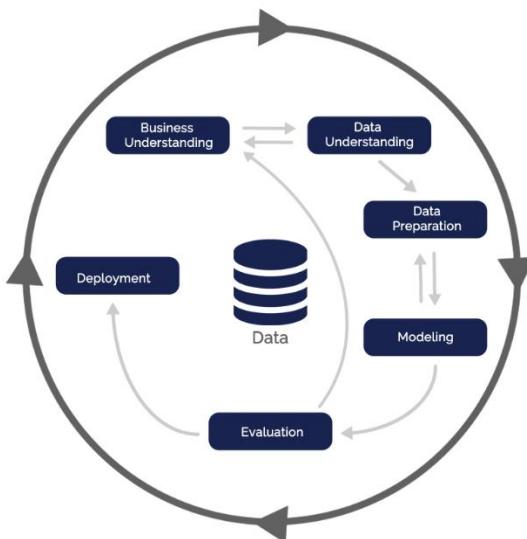
Deteksi dan pengobatan dini telah terbukti berhasil di banyak negara maju dan harus diterapkan di negara-negara dengan sumber daya terbatas di mana beberapa alat standar tersedia[6]. Angka kematian akibat kanker payudara standar usia di negara-negara berpenghasilan tinggi turun 40% antara tahun 1980-an dan 2020. Negara-negara yang berhasil menurunkan angka kematian akibat kanker payudara telah mampu mencapai penurunan angka kematian akibat kanker payudara tahunan sebesar 2-4% per tahun. Jika penurunan kematian tahunan sebesar 2,5% per tahun terjadi di seluruh dunia, 2,5 juta kematian akibat kanker payudara akan dapat dihindari antara tahun 2020 dan 2040[6].

BAB 2

Metodologi Pembuatan Aplikasi

2.1 Metodologi Pengolahan Data

Diagram alur metodologi pengolahan data yang digunakan ialah menggunakan metodologi CRISP-DM (*Cross Industry Standard Process for Data Mining*). Metodologi penelitian CRISP-DM digunakan dalam penelitian karena metodologi ini mampu menjabarkan setiap langkah pengolahan data dengan jelas serta mudah dipahami untuk diimplementasikan.



Gambar 2. 1 Metodologi CRISP-DM
Sumber: <https://www.otaris.de/en/consulting-data-science/>

Berdasarkan diagram alur metodologi penelitian diatas, dapat

dipaparkan sebagai berikut :

2.2 Business Understanding

Pada tahap ini merupakan tahap untuk menentukan seperti apa proses bisnis yang akan dibangun untuk melakukan hal apa yang ingin dicapai. Selain itu, dalam tahap ini juga diperlukan analisis apa saja alat dan bahan yang akan diperlukan untuk setiap fase kegiatan.

Dalam hal ini, tujuan kegiatan yang dilakukan adalah memprediksi kanker payudara berdasarkan faktor-faktor spesifik. Prediksi kanker tersebut akan menggunakan data kanker payudara. Pendekatan yang dilakukan dalam memprediksi kanker payudara akan dilakukan menggunakan *machine learning*.

Setelah melakukan prediksi kanker payudara menggunakan *machine learning*, model prediksi akan diterapkan dalam bentuk *web base*.

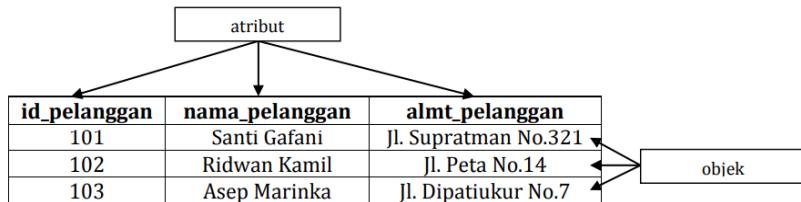
Alat yang digunakan adalah berupa perangkat keras berupa laptop. Perangkat lunak yang digunakan dalam pengolahan data kanker untuk prediksi ialah Jupyter Notebook dan Anaconda. Perangkat lunak yang digunakan dalam pembangunan web base adalah Visual Studio Code, XAMPP, dan Tableau.

2.3 Data Understanding

Pada tahap *data understanding*, hal yang dilakukan adalah melakukan pemahaman bisnis yang berfokus untuk mengidentifikasi, mengumpulkan dan menganalisis kumpulan data untuk mencapai tujuan yang dipaparkan pada *Business Understanding*.

Definisi set data (dataset/himpunan data) merupakan kumpulan objek dan atributnya. Nama lain dari objek yang sering digunakan diantaranya record, point, vector, pattern, event, observation, case, sample, instance, entitas. Objek digambarkan dengan sejumlah atribut yang menerangkan sifat atau karakteristik dari objek tersebut. Atribut juga sering disebut variabel, field, fitur, atau dimensi. Atribut adalah sifat/properti/karakteristik objek yang nilainya bisa bermacam-macam dari satu objek dengan objek

lainnya, dari satu waktu ke waktu yang lainnya. Sebagai contoh seorang pelanggan merupakan objek, dimana objek pelanggan tersebut memiliki beberapa atribut seperti id pelanggan, nama, alamat dan lain-lain. Setiap pelanggan memungkinkan memiliki nilai atribut yang berbeda dengan pelanggan lainnya, serta memungkinkan perubahan nilai atribut dari waktu ke waktu[9].



Gambar 2. 2 Struktur Dataset

Sumber: [Pertemuan 3 - Materi IDM - 2015.pdf \(unikom.ac.id\)](http://Pertemuan%203%20-%20Materi%20IDM%20-%202015.pdf)

Berikut adalah berbagai pemaparan mengenai data.

2.3.1 Tipe Data

Tipe atribut dapat dibedakan dari nilai beserta sifatnya. Ada empat sifat yang dimiliki atribut secara umum, yaitu:

1. Pembeda (distinctness): = dan ≠
2. Urutan (order): <, >, ≤, ≥
3. Penjumlahan, Pengurangan (addition): + dan -
4. Perkalian, Pembagian (multiplication): * dan /

Umumnya tipe atribut ini ada dua yaitu kategori (kualitatif) dan numerik (kuantitatif). Dari kedua tipe tersebut dibagi lagi menjadi beberapa sub tipe yang disesuaikan dengan sifat nilai yang dimilikinya[9].

2.3.2 Kualitas pada dataset

Permasalahan kualitas data ditinjau dari aspek pengukuran data

dan pengumpulan data.

1. Kesalahan Pengukuran (Measurement Error)

Kesalahan ini mengacu pada permasalahan hasil dari proses pengukuran. Problem yang umum terjadi adalah nilai yang dicatat berbeda dari nilai sebenarnya untuk beberapa tingkat. Pada atribut kontinu, beda numerik dari hasil pengukuran dengan nilai sebenarnya disebut dengan error[9]. Yang termasuk dalam jenis kesalahan pengukuran adalah sebagai berikut.

a. Noise

Merupakan komponen random dari suatu error pengukuran. Noise berkaitan dengan modifikasi dari nilai asli. Contoh: distorsi atau penyimpangan dari suara orang saat berbicara di telepon yang jaringannya buruk.

b. Bias

Suatu variasi pengukuran dari kuantitas yang sedang diukur dengan pengurangan antara mean dan nilai kuantitas yang diketahui.

c. Precision

Kedekatan dari pengukuran berulang (dari kuantitas yang sama) satu dengan yang lainnya. Diukur dengan standar deviasi.

d. Accuracy

Kedekatan pengukuran terhadap nilai sebenarnya dari kuantitas yang sedang diukur.

2. Kesalahan Pengumpulan (Collection Error)

Mengacu pada kesalahan-kesalahan (error) seperti hilangnya objek data atau nilai atribut, atau lingkup objek data yang tidak tepat[9]. Yang termasuk dalam kesalahan pengumpulan diantaranya:

a. Outliers

Merupakan objek data dengan sifat yang berbeda sekali dari kebanyakan objek data dalam data-set. Misalkan, terdapat data penelitian tentang tinggi anak siswa SMA yakni 160cm sampai 180cm. Tetapi dalam data tersebut terdapat anak yang mempunyai tinggi 140cm. Data anak dengan tinggi 140cm tersebut yang disebut data outlier, karena berbeda sangat jelas.

Terdapat beberapa hal yang mempengaruhi munculnya data outlier antara lain:

1. Kesalahan dalam pemasukan data
2. Kesalahan dalam pengambilan sample
3. Memang ada data-data ekstrim yang tidak dapat dihindarkan keberadaannya.

b. Missing value

Merupakan nilai dari suatu atribut yang tidak ditemukan. Alasan terjadinya missing value adalah:

1. Informasi tidak diperoleh (misal, orang-orang menolak untuk memberikan data umur dan berat badan)
2. Atribut yang mungkin tidak bisa diterapkan ke semua kasus (misal, pendapatan tahunan tidak bisa diterapkan pada seseorang yang pengangguran)

Penanganan missing values adalah dengan:

1. Mengurangi objek data
2. Memperkirakan missing values
3. Mengabaikan missing values pada saat analisis
4. Mengganti dengan semua nilai yang mungkin (tergantung probabilitasnya)

c. Duplicate data

Set data mungkin terdiri dari objek data yang ganda (duplikat), atau hampir selalu terjadi duplikasi antara satu dengan yang lainnya. Persoalan utama ketika menggabungkan data dari sumber-sumber yang bervariasi (heterogen).

Contoh: orang yang sama dengan alamat email yang lebih dari satu. Pembersihan data (data cleaning) merupakan proses yang berkaitan dengan permasalahan data yang duplikat[9]. []

Dalam hal ini, proses yang dilakukan ialah mengumpulkan data awal yang berasal dari Kaggle. Setelah data ditemukan, kemudian hal yang dilakukan adalah menjelajahi data dan melakukan verifikasi kualitas data (seberapa bersih atau kotornya data yang digunakan).

2.3.3 Dataset Yang Digunakan

Dalam proses melakukan pengolahan data ini, digunakan suatu material yang dapat mendukung pengolahan data yang dilakukan. Material yang dimaksud ialah berupa data yang dapat digunakan pada proses pengolahan data.

Data yang digunakan dalam penelitian merupakan data pasien kanker payudara. Data tersebut didapatkan melalui situs bernama Kaggle yang dapat dikunjungi melalui QR Code yang dicantumkan sebagai berikut:



Gambar 2. 3 Situs Download Data Kanker Payudara

Sumber: <https://www.kaggle.com/datasets/yassserh/breast-cancer-dataset>

Data bersifat berlabel dan memiliki jumlah kolom sebanyak 32 kolom dan jumlah total record sebanyak 569 record.

Kolom tersebut diantaranya adalah sebagai berikut.

Tabel 2. 1 Atribut dan Artinya dari Dataset

Atribut	Arti
id	ID unik
diagnosis	Diagnosis masalah (M = malignant/ganas, B = benign/jinak)
Radius_mean	Jari-jari lobus/kelenjar (rata-rata jarak dari pusat ke titik-titik di perimeter) Perimeter adalah garis keliling inti sel yang diukur sebagai jumlah dari jarak.
Texture_mean	Rata-rata Tekstur Permukaan (standar deviasi nilai grayscale)
Perimeter_mean	Perimeter Luar Lobus (ukuran rata-rata tumor inti)
Area_mean	Rata-rata Luas Lobus
Smoothness_mean	Rata-rata Tingkat Kehalusan (rata-rata variasi lokal dalam panjang radius)
Compactness_mean	Rata-rata dari rasio volume dan luas permukaan tumor (rata-rata dari $\text{perimeter}^2 / \text{area} - 1.0$)

Concavity_mean	Rata-rata keparahan bagian cekung dari kontur. Sebuah cekungan di bagian bawah toraks (dada) yang disebabkan oleh perpindahan tulang rawan xiphoid ke belakang.
Concave points_mean	Rata-rata untuk jumlah bagian cekung dari kontur
Symmetry_mean	Rata-rata kesimetrisan. Kesimetrisan mungkin terkait dengan penyakit payudara, masalah perkembangan, masalah hormonal, sindrom yang tidak diketahui, cedera atau periode perubahan tubuh yang signifikan seperti kehamilan atau menyusui. Asimetri payudara mengacu pada ketika satu payudara memiliki ukuran atau bentuk yang berbeda dari yang lain.
Fractal_dimension_mean	Rata-rata dimensi fraktal (rata-rata "coastline approximation" - 1) Analisis fraktal dari gambar spesimen jaringan payudara memberikan deskripsi numerik pola pertumbuhan tumor sebagai angka kontinu antara 1 dan 2. Angka ini, dimensi fraktal, adalah ukuran yang objektif dan dapat direproduksi dari kompleksitas arsitektur jaringan spesimen biopsi.

Radius_se	Jari-jari lobus/kelenjar (Standard Error dari pusat ke titik-titik di perimeter) Perimeter adalah garis keliling inti sel yang diukur sebagai jumlah dari jarak.
Texture_se	Standard Error Tekstur Permukaan (standar deviasi nilai <i>grayscale</i>)
Perimeter_se	Standard Error Perimeter Luar Lobus (ukuran Standard Error tumor inti)
Area_se	Standard Error Luas Lobus
Smoothness_se	Standard Error Tingkat Kehalusan (Standard Error variasi lokal dalam panjang radius)
Compactness_se	Standard Error dari rasio volume dan luas permukaan tumor (Standard Error dari $\text{perimeter}^2 / \text{area} - 1.0$)
Concavity_se	Standard Error keparahan bagian cekung dari kontur. Sebuah cekungan di bagian bawah toraks (dada) yang disebabkan oleh perpindahan tulang rawan xiphoid ke belakang.
Concave points_se	Standard Error untuk jumlah bagian cekung dari kontur
Symmetry_se	Standard Error kesimetrisan.

	<p>Kesimetrisan mungkin terkait dengan penyakit payudara, masalah perkembangan, masalah hormonal, sindrom yang tidak diketahui, cedera atau periode perubahan tubuh yang signifikan seperti kehamilan atau menyusui.</p> <p>Asimetri payudara mengacu pada ketika satu payudara memiliki ukuran atau bentuk yang berbeda dari yang lain.</p>
Fractal_dimension_se	<p>Standard Error dimensi fraktal (Standard Error "coastline approximation" - 1)</p> <p>Analisis fraktal dari gambar spesimen jaringan payudara memberikan deskripsi numerik pola pertumbuhan tumor sebagai angka kontinu antara 1 dan 2. Angka ini, dimensi fraktal, adalah ukuran yang objektif dan dapat direproduksi dari kompleksitas arsitektur jaringan spesimen biopsi.</p>
Radius_worst	<p>Keparahan jari-jari lobus/kelenjar (keparahan jarak dari pusat ke titik-titik di perimeter)</p> <p>Perimeter adalah garis keliling inti sel yang diukur sebagai jumlah dari jarak.</p>
Texture_worst	Keparahan Tekstur Permukaan (standar deviasi nilai <i>grayscale</i>)

Perimeter_worst	Perimeter Luar Lobus (ukuran keparahan tumor inti)
Area_worst	Keparahan Luas Lobus
Smoothness_worst	Keparahan Tingkat Kehalusan (keparahan variasi lokal dalam panjang radius)
Compactness_worst	Keparahan dari rasio volume dan luas permukaan tumor (keparahan dari $\text{perimeter}^2 / \text{area} - 1.0$)
Concavity_worst	Keparahan bagian cekung dari kontur. Sebuah cekungan di bagian bawah toraks (dada) yang disebabkan oleh perpindahan tulang rawan xiphoid ke belakang.
Concave points_worst	Keparahan untuk jumlah bagian cekung dari kontur
Symmetry_worst	Keparahan kesimetrisan. Kesimetrisan mungkin terkait dengan penyakit payudara, masalah perkembangan, masalah hormonal, sindrom yang tidak diketahui, cedera atau periode perubahan tubuh yang signifikan seperti kehamilan atau menyusui. Asimetri payudara mengacu pada ketika satu payudara memiliki ukuran atau bentuk yang berbeda dari yang lain.
Fractal_dimension_worst	Keparahan dimensi fraktal (rata-rata "coastline approximation" - 1)

	Analisis fraktal dari gambar spesimen jaringan payudara memberikan deskripsi numerik pola pertumbuhan tumor sebagai angka kontinu antara 1 dan 2. Angka ini, dimensi fraktal, adalah ukuran yang objektif dan dapat direproduksi dari kompleksitas arsitektur jaringan spesimen biopsi.
--	---

Berdasarkan data tersebut, berikut adalah pemaparan kode python pada Jupyter Notebook dalam tahap *Data Understanding*.

```
[1] import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns
    import numpy as np
```

```
[2] df = pd.read_csv('breast-cancer.csv')
df.head()
```

Output:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave_points_mean	..._mean	rac
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	...	
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	...	
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	...	
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	...	
4	84358402	M	20.29	14.34	135.10	1257.0	0.10030	0.13280	0.1980	0.10430	...	

5 rows × 32 columns

```
[3] df.info()
```

Output:

```

Data columns (total 32 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   id               569 non-null     int64  
 1   diagnosis        569 non-null     object  
 2   radius_mean      569 non-null     float64 
 3   texture_mean     569 non-null     float64 
 4   perimeter_mean   569 non-null     float64 
 5   area_mean         569 non-null     float64 
 6   smoothness_mean  569 non-null     float64 
 7   compactness_mean 569 non-null     float64 
 8   concavity_mean   569 non-null     float64 
 9   concave_points_mean 569 non-null   float64 
 10  symmetry_mean   569 non-null     float64 
 11  fractal_dimension_mean 569 non-null   float64 
 12  radius_se        569 non-null     float64 
 13  texture_se       569 non-null     float64 
 14  perimeter_se    569 non-null     float64 
 15  area_se          569 non-null     float64 
 16  smoothness_se   569 non-null     float64 
 17  compactness_se  569 non-null     float64 
 18  concavity_se    569 non-null     float64 
 19  concave_points_se 569 non-null   float64 
 20  symmetry_se     569 non-null     float64 
 21  fractal_dimension_se 569 non-null   float64 
 22  radius_worst    569 non-null     float64 
 23  texture_worst   569 non-null     float64 
 24  perimeter_worst 569 non-null     float64 
 25  area_worst       569 non-null     float64 
 26  smoothness_worst 569 non-null     float64 
 27  compactness_worst 569 non-null   float64 
 28  concavity_worst 569 non-null     float64 
 29  concave_points_worst 569 non-null   float64 
 30  symmetry_worst  569 non-null     float64 
 31  fractal_dimension_worst 569 non-null   float64 
dtypes: float64(30), int64(1), object(1)
memory usage: 142.4+ KB

```

[4]

df.columns

Output:

```

Index(['id', 'diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean',
       'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean',
       'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean',
       'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',
       'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se',
       'fractal_dimension_se', 'radius_worst', 'texture_worst',
       'perimeter_worst', 'area_worst', 'smoothness_worst',
       'compactness_worst', 'concavity_worst', 'concave points_worst',
       'symmetry_worst', 'fractal_dimension_worst'],
      dtype='object')

```

[5]

df = df.drop(['id'],axis=1)

Keterangan :

- Baris 1 : Mengimport seluruh library yang dibutuhkan.
- Baris 2 : Mengimport data kedalam variabel dengan nama “df”. Kemudian menampilkan 5 data teratas.
- Baris 3 : Menampilkan informasi detail tabel dan data.

- d. Baris 4 : Menampilkan kolom apa saja yang terdapat di dalam data.
- e. Baris 5 : Menghapus atribut “id“, karena tidak akan terpakai.

2.4 Data Preparation

Tahap *data preparation* dilakukan untuk menyiapkan data akhir untuk dilakukannya pemodelan dengan cara memperbaiki masalah yang terdapat pada data. Hal yang dilakukan dalam fase ini adalah membersihkan data, serta memilih data yang akan digunakan dalam pemodelan. Sehingga nantinya akan didapatkan data akhir yang siap dimodelkan.

Adapun tahap *data preparation* pada fase ini adalah sebagai berikut.

```
[6] df.isnull().any()
```

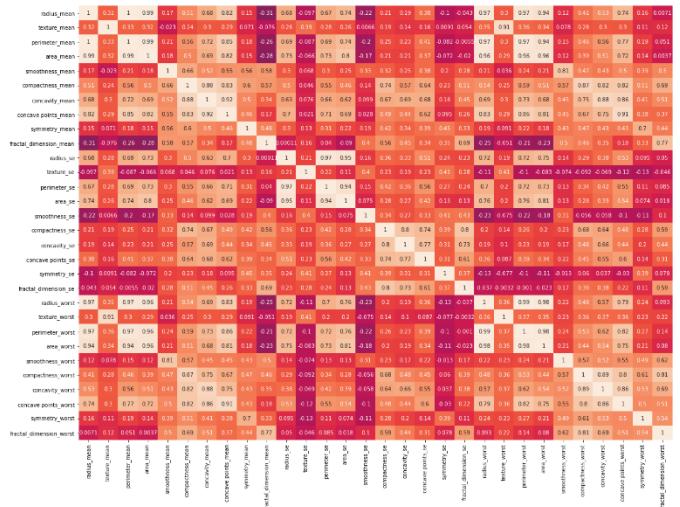
Output:

```
diagnosis          False
radius_mean        False
texture_mean       False
perimeter_mean    False
area_mean          False
smoothness_mean   False
compactness_mean  False
concavity_mean    False
concave_points_mean False
symmetry_mean     False
fractal_dimension_mean False
radius_se          False
texture_se         False
perimeter_se       False
area_se            False
smoothness_se     False
compactness_se    False
concavity_se      False
concave_points_se False
symmetry_se       False
fractal_dimension_se False
radius_worst       False
texture_worst      False
perimeter_worst   False
area_worst         False
smoothness_worst  False
compactness_worst False
concavity_worst   False
concave_points_worst False
symmetry_worst    False
fractal_dimension_worst False
dtype: bool
```

[7]

```
plt.figure(figsize=(25, 15))
heatmap = sns.heatmap(df.corr(), vmin=-1, vmax=1, annot=True)
heatmap.set_title('Correlation Heatmap', fontdict={'fontsize':4}, pad=12);
```

Output:



radius_mean texture_mean perimeter_mean area_mean smoothness_mean compactness_mean concavity_mean concave points_mean symmetry_mean fractal_dimension_mean
radius_worst texture_worst perimeter_worst area_worst smoothness_worst compactness_worst concavity_worst concave points_worst symmetry_worst fractal_dimension_worst

[8]

```
df.describe()
```

Output:

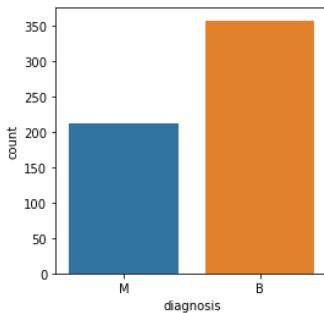
	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symmetry_mean	fract
count	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000
mean	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.088799	0.048919	0.181162	
std	3.524049	4.301036	24.296981	351.914129	0.014064	0.052613	0.079720	0.038603	0.027414	
min	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	0.106000	
25%	11.700000	16.170000	75.170000	240.300000	0.086370	0.064920	0.029560	0.020310	0.161900	
50%	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	0.179200	
75%	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	0.195700	
max	28.110000	39.280000	188.500000	2501.000000	0.163400	0.345400	0.426800	0.201200	0.304000	

8 rows × 30 columns

[9]

```
plt.figure(figsize=(4,4))
sns.countplot(data = df,x = 'diagnosis')
```

Output:



[10] df = df.drop(['texture_mean', 'smoothness_mean', 'symmetry_mean',
 'fractal_dimension_mean', 'area_se', 'smoothness_se',
 'symmetry_se', 'smoothness_worst', 'symmetry_worst',
 'fractal_dimension_worst'], axis=1)

[11] df.rename(columns = {'radius_mean': 'rerata_jari2_lobus',
 'perimeter_mean' : 'rerata_tumor_inti',
 'area_mean' : 'rerata_luas_lobus',
 'compactness_mean' : 'rerata_luas_permukaan_tumor',
 'concavity_mean' : 'rerata_cekungan_kontur',
 'concave points_mean': 'rerata_jumlah_cekungan_kontur',
 'radius_se' : 'se_jari2_lobus',
 'texture_se' : 'se_tekstur_permukaan',
 'perimeter_se' : 'se_tumor_inti',
 'compactness_se' : 'se_luas_permukaan_tumor',
 'concavity_se' : 'se_cekungan_kontur',
 'concave points_se' : 'se_jumlah_cekungan_kontur',
 'fractal_dimension_se': 'se_fraktal_spesimen',
 'radius_worst' : 'keparahan_jari2_lobus',
 'texture_worst' : 'keparahan_tekstur_permukaan' ,
 'perimeter_worst' : 'keparahan_tumor_inti',
 'area_worst' : 'keparahan_luas_lobus',
 'compactness_worst' : 'keparahan_luas_permukaan_tumor',
 'concavity_worst' : 'keparahan_cekungan_kontur',
 'concave points_worst' : 'keparahan_jumlah_cekungan_kontur'},
 inplace = True)
 df.info()

Output:

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 21 columns):
 #   Column           Non-Null Count Dtype  
 --- 
 0   diagnosis        569 non-null   object  
 1   rerata_jari2_lobus 569 non-null   float64 
 2   rerata_tumor_inti  569 non-null   float64 
 3   rerata_luas_lobus  569 non-null   float64 
 4   rerata_luas_permukaan_tumor 569 non-null   float64 
 5   rerata_cekungan_kontur 569 non-null   float64 
 6   rerata_jumlah_cekungan_kontur 569 non-null   float64 
 7   se_jariz_lobus     569 non-null   float64 
 8   se_tekstur_permukaan 569 non-null   float64 
 9   se_tumor_inti     569 non-null   float64 
 10  se_luas_permukaan_tumor 569 non-null   float64 
 11  se_cekungan_kontur 569 non-null   float64 
 12  se_jumlah_cekungan_kontur 569 non-null   float64 
 13  se_fraktal_spesimen 569 non-null   float64 
 14  keparahan_jariz_lobus 569 non-null   float64 
 15  keparahan_tekstur_permukaan 569 non-null   float64 
 16  keparahan_tumor_inti 569 non-null   float64 
 17  keparahan_luas_lobus 569 non-null   float64 
 18  keparahan_luas_permukaan_tumor 569 non-null   float64 
 19  keparahan_cekungan_kontur 569 non-null   float64 
 20  keparahan_jumlah_cekungan_kontur 569 non-null   float64 
dtypes: float64(20), object(1)
memory usage: 93.5+ KB

```

[12]

```

from sklearn.preprocessing import StandardScaler
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
df = df.apply(le.fit_transform)
df

```

Output:

	diagnosis	rerata_jari2_lobus	rerata_tumor_inti	rerata_luas_lobus	rerata_luas_permukaan_tumor	rerata_cekungan_kontur	rerata_jumlah_cekungan_kontur	s
0	1	370	455	452		531	521	526
1	1	426	486	513		215	321	396
2	1	406	478	487		459	478	516
3	1	98	164	95		533	507	493
4	1	418	493	506		414	480	492
...
564	1	438	503	522		364	509	522
565	1	413	483	499		312	415	483
566	1	340	410	421		308	329	335
567	1	429	500	501		530	530	531
568	0	3	1	3		37	0	0

569 rows × 21 columns

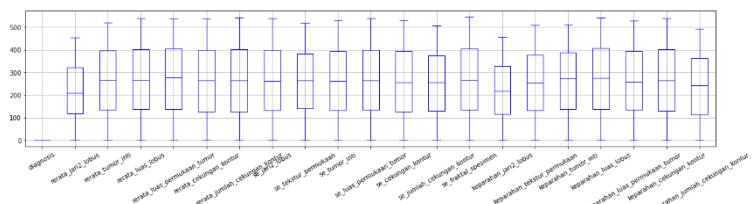
[13]

```

plt.figure(figsize=(20,4))
df.boxplot(grid='false', color='blue', fontsize=10, rot=30 )

```

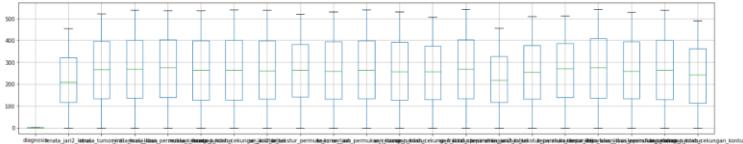
Output:



```
[14] plt.rcParams["figure.figsize"] = [20,4]
plt.rcParams["figure.autolayout"] = True

_, bp = pd.DataFrame.boxplot(df, return_type='both')
outliers = [flier.get_ydata() for flier in bp["fliers"]]

print("Outliers: ", outliers)
```



```
[15] # Hitung nilai Q1 dan Q3
Q1 = np.quantile(df, .25)
Q3 = np.quantile(df, .75)
# Hitung nilai IQR
IQR = Q3 - Q1
print('Kuartil 1 = ', Q1)
print('Kuartil 3 = ', Q3)
print('IQR = ', IQR)
```

Output:

Kuartil 1 = 110.0
Kuartil 3 = 379.0
IQR = 269.0

```
[16] min_IQR = Q1 - 1.5 * IQR  
max_IQR = Q3 + 1.5 * IQR  
  
print('IQR minimum = ', min_IQR)  
print('IQR maksimum = ', max_IQR)
```

Output:

IQR minimum = -293.5
IQR maksimum = 782.5

Keterangan :

- a. Baris 6 : Menampilkan apakah terdapat data kosong pada

suatu atribut atau tidak.

- b. Baris 7 : Membuat dan menampilkan korelasi antar atribut menggunakan heatmap correlation yang berasal dari library seaborn.
- c. Baris 8 : Menampilkan hasil perhitungan dari statistika berupa count, mean, std, min, 25%, 50%, 75%, max.
- d. Baris 9 : Membuat dan menampilkan Bar Chart dari atribut “diagnosis” untuk mengetahui jumlah antara Malignant (M) dan Benign (B).
- e. Baris 10 : Menghapus atribut yang tidak terpakai, karena tidak memiliki korelasi antar atribut yang kuat
- f. Baris 11 : Mengubah seluruh nama atribut menjadi nama yang lebih mudah dipahami
- g. Baris 12 : Mengubah skala atribut menggunakan library StandardScaler dan preprocessing. Atribut “diagnosis” memiliki nilai yang berubah menjadi angka (M menjadi 1, B menjadi 0).
- h. Baris 13 : Membuat dan menampilkan boxplot untuk melihat outlier pada data.
- i. Baris 14 : Membuat dan menampilkan boxplot untuk melihat outlier pada data dan menampilkan hasil outlier dengan array.
- j. Baris 15 : Menghitung nilai Q1 dan Q3 untuk menampilkan hasil Kuartil 1, Kuartil 2, dan nilai atribut.
- k. Baris 16 : menampilkan hasil IQR maksimum dan minimum

2.5 *Modelling*

Tahap *modelling* atau pemodelan adalah menentukan algoritma teknik pemodelan yang akan digunakan untuk menyelesaikan masalah yang ada. Dalam fase ini, model yang digunakan adalah menggunakan model *Support Vector Machine*.

Karena model ini adalah model yang cocok untuk digunakan dalam melakukan prediksi terutama jika variabel yang mempengaruhi sangat banyak.

2.5.1 Support Vector Machine

SVM adalah suatu aturan algoritma *machine learning* berupa *supervised learning* yang digunakan untuk setiap klasifikasi dan tantangan regresi. Dimanfaatkan dalam masalah klasifikasi, dalam algoritma SVM, mengarah untuk memplot setiap item pengetahuan sebagai batas tertentu dalam ruang n-dimensi (dimana n adalah banyak fitur yang dimiliki) dengan nilai setiap fitur menjadi nilai koordinat tertentu. Selanjutnya, dikategorikan dengan menemukan level super yang membedakan 2 kategori (kelas) dengan sangat baik[10].

Teknik ini menarik orang dalam bidang data mining maupun machine learning karena performansinya yang meyakinkan dalam memprediksi kelas suatu data baru.

Pembahasan awal adalah pembahasan dengan kasus klasifikasi yang secara linier bisa dipisahkan. Dalam hal ini fungsi pemisah yang kita cari adalah fungsi linier[10]. Fungsi ini bisa didefinisikan sebagai :

Dengan syarat :

$$y_i(x_i \cdot w + b) - 1 \geq 0, i = 1, 2, 3, \dots, n$$

$$y_i(x_1.w_1 + x_2.w_2 + b) \geq 1$$

Dalam teknik ini kita berusaha menemukan fungsi pemisah (klasifier/hyperplane) terbaik diantara fungsi yang tidak terbatas jumlahnya untuk memisahkan dua macam obyek. Hyperplane terbaik adalah hyperplane yang terletak di tengah-tengah antara dua set obyek dari dua kelas. Mencari hyperplane terbaik ekuivalen dengan memaksimalkan margin atau jarak antara dua set obyek dari kelas yang berbeda. Jika $wx_1 + b = +1$ adalah

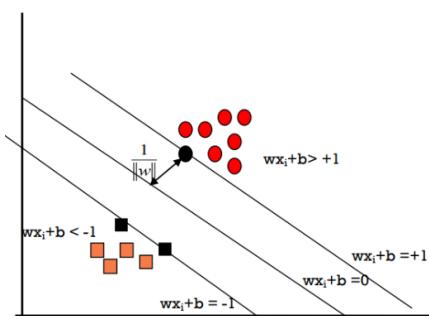
hyperplane-pendukung (*supporting hyperplane*) dari kelas +1, dan $wx_2 + b = -1$ adalah hyperplane-pendukung dari kelas -1, margin antara dua kelas dapat dihitung dengan mencari jarak antara kedua hyperplane-pendukung dari kedua kelas. Secara spesifik, margin dihitung dengan cara berikut :

$$w(x_1 - x_2) = 2 \dots \dots \dots \dots \dots \dots \quad (5)$$

Menjadi:

$$\frac{w(x_1 - x_2)}{\|w\|} = \frac{2}{\|w\|} \dots \dots \dots (6)$$

Untuk membuktikan bahwa memaksimalkan margin antara dua set obyek akan meningkatkan probabilitas pengelompokan secara benar dari data testing. Pada dasarnya jumlah fungsi pemisah ini tidak terbatas banyaknya. Misalkan dari jumlah yang tidak terbatas ini di ambil dua saja, yaitu $f_1(x)$ and $f_2(x)$. Fungsi f_1 mempunyai margin yang lebih besar dari pada fungsi f_2 . Setelah menemukan dua fungsi ini, suatu data baru masuk dengan keluaran -1 . Data ini harus dikelompokkan apakah data ini ada dalam kelas -1 atau $+1$ menggunakan fungsi pemisah yang sudah kita temukan.



Gambar 2-4 Visualisasi SVM

Sumber : <https://medium.com/@samsudiney/penjelasan-sederhana-tentang-apa itu-svm-149fec72bd02>

Mencari fungsi pemisah yang optimal untuk obyek yang bisa dipisahkan secara linier Dengan menggunakan f1, kita akan kelompokkan data baru ini di kelas -1 yang berarti kita benar mengelompokkannya. Sekarang kita coba gunakan f2, kita akan menempatkannya di kelas +1 yang berarti salah. Dari contoh sederhana ini kita lihat bahwa memperbesar margin bisa meningkatkan probabilitas pengelompokan suatu data secara benar[10].

Adapun tahap *modelling* pada fase ini adalah sebagai berikut.

```
[17] Diagnosis = pd.get_dummies(df['diagnosis'],drop_first=True)
Diagnosis.head()
```

Output:

	1
0	1
1	1
2	1
3	1
4	1

```
[18] Diagnosis.rename(columns = {1 : 'diagnosis'}, inplace = True)
Diagnosis.head()
```

Output:

	diagnosis
0	1
1	1
2	1
3	1
4	1

```
[19] Diagnosis['diagnosis'].replace([1, 0],
                                    ['Tumor Ganas (Terdeteksi Kanker)',
                                     'Tumor Jinak (Tidak Terdeteksi Kanker)'],
                                    inplace=True)
```

Output:

diagnosis	
0	Tumor Ganas (Terdeteksi Kanker)
1	Tumor Ganas (Terdeteksi Kanker)
2	Tumor Ganas (Terdeteksi Kanker)
3	Tumor Ganas (Terdeteksi Kanker)
4	Tumor Ganas (Terdeteksi Kanker)

[20] df.drop(['diagnosis'],axis=1,inplace=True)

[21] df = pd.concat([df,Diagnosis],axis=1)
df.head()

Output:

	rerata_jari2_libus	rerata_tumor_inti	rerata_luas_libus	rerata_luas_permukaan_tumor	rerata_cekungan_kontur	rerata_jumlah_cekungan_kontur	se_jari2_libus
0	370	455	452	531	521	526	525
1	426	486	513	215	321	396	424
2	406	478	487	459	478	516	481
3	98	164	95	533	507	493	404
4	418	493	506	414	480	492	484

5 rows × 21 columns

[22] y = Diagnosis
y.head()

Output:

diagnosis	
0	Tumor Ganas (Terdeteksi Kanker)
1	Tumor Ganas (Terdeteksi Kanker)
2	Tumor Ganas (Terdeteksi Kanker)
3	Tumor Ganas (Terdeteksi Kanker)
4	Tumor Ganas (Terdeteksi Kanker)

[23] X=df.drop(['diagnosis'],axis=1)
X.head()

Output:

	rerata_jari2_lobus	rerata_tumor_inti	rerata_luas_lobus	rerata_luas_permukaan_tumor	rerata_cekungan_kontur	rerata_jumlah_cekungan_kontur	se_jari2_lobus
0	370	455	452	531	521	526	525
1	426	486	513	215	321	396	424
2	406	478	487	459	478	516	481
3	98	164	95	533	507	493	404
4	418	493	506	414	480	492	484

[24] `from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
 X, y, test_size=0.2, random_state=101)`

[25] `from sklearn.svm import SVC
model = SVC()
model.fit(X_train,y_train)
persamaan = model.fit(X_train,y_train)`

Output:

SVC()

[26] `pred = model.predict(X_test)
pred`

Output:

```
array(['Tumor Jinak (Tidak Terdeteksi Kanker)',  
       'Tumor Jinak (Tidak Terdeteksi Kanker)',  
       'Tumor Jinak (Tidak Terdeteksi Kanker)',  
       'Tumor Ganas (Terdeteksi Kanker)',  
       'Tumor Jinak (Tidak Terdeteksi Kanker)',  
       'Tumor Jinak (Tidak Terdeteksi Kanker)',  
       'Tumor Jinak (Tidak Terdeteksi Kanker)',  
       'Tumor Ganas (Terdeteksi Kanker)',  
       'Tumor Jinak (Tidak Terdeteksi Kanker)',  
       'Tumor Jinak (Tidak Terdeteksi Kanker)',  
       'Tumor Ganas (Terdeteksi Kanker)',  
       'Tumor Jinak (Tidak Terdeteksi Kanker)',  
       'Tumor Jinak (Tidak Terdeteksi Kanker)',  
       'Tumor Jinak (Tidak Terdeteksi Kanker)',  
       'Tumor Ganas (Terdeteksi Kanker)',  
       'Tumor Jinak (Tidak Terdeteksi Kanker)',  
       'Tumor Jinak (Tidak Terdeteksi Kanker)',  
       'Tumor Jinak (Tidak Terdeteksi Kanker)',  
       'Tumor Ganas (Terdeteksi Kanker)',  
       'Tumor Jinak (Tidak Terdeteksi Kanker)',  
       'Tumor Jinak (Tidak Terdeteksi Kanker)',  
       'Tumor Jinak (Tidak Terdeteksi Kanker)',  
       'Tumor Ganas (Terdeteksi Kanker)'])
```

Keterangan :

- a. Baris 17 : Menyimpan dataframe menjadi dummy pada atribut “diagnosis” pada variabel “Diagnosis“. Ketika ditampilkan 5 data teratas, atribut berubah menjadi “1“
- b. Baris 18 : Merubah atribut dataframe “Diagnosis“ yang awalnya “1“ menjadi “diagnosis“

- c. Baris 19 : Mengubah seluruh nilai dataframe “Diagnosis” dimana nilai “1“ menjadi “Tumor Ganas (Terdeteksi Kanker)“ dan nilai “0“ menjadi “Tumor Jinak (Tidak Terdeteksi Kanker)“
- d. Baris 20 : Menghapus atribut “diagnosis“ pada df
- e. Baris 21 : Menggabungkan dataframe “df“ dan dataframe “Diagnosis“
- f. Baris 22 : Menyimpan dataframe “Diagnosis“ pada variabel “y“
- g. Baris 23 : Menghapus atribut “diagnosis“ pada df, kemudian disimpan dalam variabel “X“
- h. Baris 24 : Melakukan split data menjadi data training (sebanyak 80%) dan data testing (sebanyak 20%) menggunakan library train_test_split
- i. Baris 25 : Membuat model menggunakan Support Vector Machine dan membuat ukuran kesesuaian model menggunakan library SVC. Kemudian model disimpan kedalam variabel “Persamaan“
- j. Baris 26 : Melakukan cek hasil prediksi menggunakan dataset testing X_test

2.6 Evaluation

Tahap *evaluation* lebih berfokus untuk melihat apakah pemodelan yang dilakukan memiliki hasil yang baik atau malah sebaliknya. Pada tahap ini, evaluasi model dilakukan dengan menggunakan metode akurasi dan *confussion matrix* untuk melihat hasil-hasil evaluasi dari model.

Adapun tahap *evaluation* pada fase ini adalah sebagai berikut.

```
[26] from sklearn.metrics import classification_report,confusion_matrix
print(classification_report(y_test,pred))
print('\n')
print(confusion_matrix(y_test,pred))
```

Output:

	precision	recall	f1-score	support
Tumor Ganas (Terdeteksi Kanker)	1.00	0.95	0.98	42
Tumor Jinak (Tidak Terdeteksi Kanker)	0.97	1.00	0.99	72
accuracy			0.98	114
macro avg	0.99	0.98	0.98	114
weighted avg	0.98	0.98	0.98	114

```
[[40  2]
 [ 0 72]]
```

[27]

```
cm = confusion_matrix(y_test, pred)
sns.heatmap(cm, annot=True)
print(classification_report(y_test,pred))
```

Output:

	precision	recall	f1-score	support
Tumor Ganas (Terdeteksi Kanker)	1.00	0.95	0.98	42
Tumor Jinak (Tidak Terdeteksi Kanker)	0.97	1.00	0.99	72
accuracy			0.98	114
macro avg	0.99	0.98	0.98	114
weighted avg	0.98	0.98	0.98	114



[28]

```
from sklearn.metrics import accuracy_score,
confusion_matrix,precision_score,f1_score,recall_score
print("accuracy score:", (accuracy_score(pred, y_test))*100, "%")
```

Output:

```
accuracy score: 98.24561403508771 %
```

[29]

```
print(model.predict([[370,200,455,452,444,531,521,526,416,483,
370,200,455,452,444,531,521,526,416,483]]))
```

Output:

```
['Tumor Ganas (Terdeteksi Kanker)']
```

Keterangan :

- a. Baris 26 : Membuat dan menampilkan confusion matrix untuk mengetahui akurasi model.
- b. Baris 27 : Membuat dan menampilkan confusion matrix untuk mengetahui akurasi model menggunakan seaborn agar lebih mudah dibaca.
- c. Baris 28 : Menampilkan hasil akurasi model prediksi yang dibuat menggunakan library accuracy_score
- d. Baris 29 : Melakukan percobaan prediksi dengan inputan manual.

2.7 Deployment

Yang terakhir ialah tahap *deployment*. Dimana pada tahap ini, hasil model penelitian yang telah dilakukan akan dikembangkan dan didokumentasikan. Pada tahap ini, hal yang dilakukan dalam fase ini yaitu mengembangkan model ke dalam bentuk aplikasi sehingga lebih mudah digunakan dalam melakukan proses prediksi.

Adapun tahap *deployment* pada fase ini adalah sebagai berikut.

[30] `import pickle`

[31] `pd.to_pickle(persamaan, "model.pickle")`

Keterangan :

- a. Baris 30 : Mengimport library pickle untuk menyimpan data
- b. Baris 31 : Menyimpan model (variabel “persamaan”) kedalam bentuk pickle, dan memberi nama file menjadi “model.pickle”

BAB 3

Apa Saja Perangkat Yang Digunakan Untuk Membuat Aplikasi?

3.1 Bahasa Pemrograman dan Pendukung Yang Digunakan

3.1.1 Python



Gambar 3. 1 Logo Python
Sumber : <https://www.python.org/community/logos/>

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Pyhton disebutkan sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standat yang besar serta komprehensif. Python mendukung multi paradigma pemrograman, utamanya; namun tidak dibatasi; pada pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional. Salah satu fitur yang tersedia pada python adalah sebagai bahasa pemrograman dinamis yang dilengkapi dengan manajemen memori otomatis.

Untuk mendapatkan bahasa pemrograman Python, dapat mengunjungi situs resmi pada QR Code berikut.



Gambar 2. 5 Situs Download Python
Sumber : <https://www.python.org/downloads/>

Seperti halnya bahasa pemrograman dinamis lainnya, python umumnya digunakan sebagai bahasa *script* meski pada praktiknya penggunaan bahasa ini lebih luas mencakup konteks pemanfaatan yang umumnya tidak dilakukan dengan menggunakan bahasa *script*. Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi. Saat ini kode python dapat dijalankan di berbagai platform sistem operasi, beberapa diantaranya adalah :

- a. Linux/Unix
- b. Windows
- c. Max OS X
- d. Java Virtual Machine
- e. Amiga
- f. Palm
- g. Symbian (untuk produk-produk Nokia)

Python didistribusikan dengan beberapa lisensi yang berbeda dari beberapa versi. Namun pada prinsipnya Python dapat diperoleh dan dipergunakan secara bebas, bahkan untuk kepentingan komersial. Lisensi Python tidak bertentangan baik menurut definisi Open Source maupun General Public License (GPL).

Bahasa pemrograman Python tentunya tidak dapat diimplementasikan atau digunakan secara langsung pada *software* teks editor ataupun terminal pada *device* seperti bahasa pemrograman PHP. Bahasa pemrograman ini akan dapat digunakan jika telah ter-install pada *device* yang digunakan.

3.1.2 HTML



Gambar 3. 2 Logo HTML

Sumber : https://www.kindpng.com/imgv/iJJbTR_java-html-language-logo-png-transparent-png/

HTML (*Hyper Text Mark Up Language*) ialah bahasa yang dipergunakan dengan tujuan untuk mendeskripsikan struktur dari sebuah halaman web. HTML memiliki fungsi untuk mempublikasikan dokumen secara online. *Tags* adalah statement dasar dari HTML. Sebuah *tag* dinyatakan dalam kurung siku (<>). *Tags* yang ditujukan untuk sebuah dokumen atau bagian dari suatu dokumen harus dibuat dalam bentuk berpasangan. *Tag* terdiri dari dua macam yaitu *tag* pembuka dan *tag* penutup. Dimana *tag* penutup menggunakan tambahan tanda garis miring (/) di awal nama *tag*[11].

3.1.3 CSS



Gambar 3. 3 Logo CSS

Sumber :

https://commons.wikimedia.org/wiki/File:CSS3_logo_and_wordmark.svg

CSS memiliki kepanjangan *Cascading Style Sheet* adalah bahasa-bahasa yang dapat merepresentasikan halaman web. Representasi tersebut diantaranya adalah warna, *layout*, dan *font*. Dengan digunakannya CSS dalam membangun halaman web, seorang web developer dapat membuat halaman web yang dapat beradaptasi dengan berbagai macam ukuran layar. Pembuatan skrip CSS biasanya terpisah dengan halaman HTML. Meskipun skrip CSS dapat disisipkan di dalam halaman HTML. Hal ini ditujukan untuk memudahkan pengaturan halaman HTML yang memiliki rancangan yang sama [11].

3.1.4 Bootstrap



Gambar 3. 4 Logo Bootstrap

Sumber : <https://pkp.sfu.ca/2016/09/02/bootstrap-theme-for-ojs-3/>

Bootstrap adalah suatu *framework* untuk CSS (*Cascading Style Sheet*) dan berupa produk bentuk opensource (terbuka untuk siapapun) yang dibuat oleh Mark Otto dan Jacob Thornton. Pada awalnya Bootstrap ini dibuat untuk membuat standarisasi dalam front end untuk seluruh programmer yang ada di perusahaannya. Bootstrap telah berubah dari yang sebelumnya adalah CSS-Driven proyek ke sebuah host dari JavaScript plugins dan ikon yang dapat dengan mudah digunakan untuk formulir dan tombol. Untuk mengunduh folder dari Bootstrap dapat dilakukan scanning pada QR Code berikut.



Gambar 3. 5 Situs Download Bootstrap

Sumber : <https://getbootstrap.com/docs/5.0/getting-started/download/>

3.1.5 Django



Gambar 3. 6 Logo Django

Sumber : <https://www.djangoproject.com/community/logos/>

Django ialah sebuah web *framework* berbasis bahasa pemrograman Python yang didesain untuk membuat suatu aplikasi web yang dinamis, kaya fitur dan aman. [12] Django yang dikembangkan oleh Django *Software Foundation* terus mendapatkan perbaikan sehingga membuat web *framework* yang satu ini menjadi pilihan utama bagi banyak pengembang aplikasi web. [12]

Untuk menginstal Framework Django dapat dilakukan perintah berikut.

```
$ python -m pip install Django
```

Atau mengunjungi situs resmi dari Django dapat dilakukan scanning pada QR Code berikut.



Gambar 3. 7 Situs Donwload Django

Sumber : <https://docs.djangoproject.com/en/4.1/topics/install/>

Framework Django bertujuan untuk memudahkan pengembangan situs web dan basis data yang kompleks (Django Software Foundation, 2017). Salah satu keunggulan django yaitu *framework* ini telah merepresentasikan model ORM (*Object Relational Mapper*) sehingga tidak perlu lagi untuk menyesuaikan *query* jika terjadi perubahan *database* yang digunakan. Berikut ini adalah beberapa keuntungan menggunakan Django:

1. *Object-Relational Mapping (ORM) Support.*
2. *Automatic Admin Interface.*
3. *Elegant URL Design.*
4. *Template System.*
5. *Cache system.*
6. *Internationalization.*
7. *A light weight web server for development and test*[13].

3.2 Penyimpanan Yang Digunakan

3.2.1 Database MySQL



Gambar 3. 8 Logo MySQL
Sumber : <https://www.logo.wine/logo/MySQL>

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (Bahasa Inggris: *database management system*) atau

DBMS yang *multihead*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL adalah DBMS yang *open source* dengan dua bentuk lisensi, yaitu *Free Software* (perangkat lunak bebas) dan *Shareware* (perangkat lunak berpemilik yang penggunaannya terbatas). Jadi MySQL adalah *database server* yang gratis dengan lisensi GNU *General Public License* (GPL) sehingga dapat dipakai untuk keperluan pribadi atau komersil tanpa harus membayar lisensi yang ada. Istilah semacam baris, kolom, tabel, dipakai pada MySQL. SQL sendiri merupakan suatu Bahasa yang dipakai di dalam pengambilan data pada *relational database* atau *database* yang terstruktur.

Sehingga MySQL adalah *database management system* yang menggunakan Bahasa SQL sebagai Bahasa penghubung antara perangkat lunak aplikasi dengan *database server*. MySQL sendiri sebenarnya merupakan lanjutan dari proyek UNIREG yang dikerjakan oleh Michael Montry Widenius dan TcX (perusahaan perangkat lunak asal Swedia). Sayangnya, UNIREG belum terlalu kompatibel dengan *database* dinamis yang dipakai di *website*. TcX kemudian mencari alternatif lain dan menemukan perangkat lunak yang dikembangkan oleh David Hughes, yaitu miniSQL atau mSQL. Namun, ditemukan masalah lagi karena mSQL tidak mendukung *indexing* sehingga belum sesuai dengan kebutuhan TcX. MySQL *server* mendukung beberapa macam sistem operasi (*cross-platform*), khususnya Linux dan Windows. Proses instalasinya pun sangat mudah. Jika menggunakan Windows, bisa menginstall XAMPP untuk menjalankan MySQL *server* yang di dalamnya sudah terdapat juga modul untuk menjalankan Apache, PHP, FileZilla, dan Tomcat. Sedangkan di Linux, dapat menginstall MySQL secara terpisah atau menginstall LAMP (Linux, Apache, MySQL, PHP) yang sudah ada pada modul Apache dan PHP. Selain itu, XAMPP juga sudah tersedia di Linux[14].

3.3 Perangkat Lunak Yang Dibutuhkan

3.3.1 Visual Studio Code



Gambar 3. 9 Logo Visual Studio Code
Sumber : <https://www.youtube.com/c/code>

Visual Studio Code (VS Code) adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi Linux, Mac, dan Windows. Teks editor ini secara langsung mendukung bahasa pemrograman dengan bantuan plugin yang dapat dipasang via *marketplace* Visual Studio Code[15].

Untuk mengunduh *software* dari Visual Studio Code dapat dilakukan scanning pada QR Code berikut.



Gambar 3. 10 Situs Download Visual Studio Code
Sumber : <https://code.visualstudio.com/>

Berikut fitur-fitur unggulan Visual Studio Code :

1. *Cross platform*

Dapat berjalan di Windows, Linux, dan MacOS. Artinya dapat bekerja pada sistem operasi manapun dengan menggunakan Visual Studio Code.

2. *Lightweight*

Tak perlu menunggu lama untuk memulai. Bahasa, tema, debugger, commands, dan lain-lainnya dapat dikontrol sepenuhnya sesuai keinginan.

3. *Powerfull Editor*

Source code editing yang sangat produktif, seperti membuat code *snippets*, *IntelliSense*, *auto correct*, dan *formatting*.

4. *Code Debugging*

Salah satu fitur bagus yang ditawarkan Visual Studio Code adalah membantu untuk melakukan debug pada kode dengan cara mengawasi kode, variabel, *call stack* dan *expression* yang mana saja.

5. *Source Control*

Visual Studio Code memiliki *intgrated source control* termasuk *Git support in-the-box* dan penyedia *source code control* lainnya di pasaran. Ini meningkatkan siklus rilis proyek secara signifikan.

6. *Intergrated Terminal*

Tidak ada lagi *multiple Windows* dan Alt-Tabs. Perintah tersebut dapat dilakukan dengan *command-line task* sekejap dan membuat banyak terminal di dalam editor.

3.3.2 XAMPP



Gambar 3. 11 Logo XAMPP

Sumber : <https://www.ekrut.com/media/xampp-adalah>

XAMPP adalah sebuah perangkat lunak komputer yang sistem penamaannya diambil dari akronim kata Apache, MySQL/MariaDB, PHP, dan Perl. Sementara imbuhan huruf “X” yang terdapat pada awal kata berasal dari istilah *cross platform* sebagai simbol bahwa aplikasi ini bisa dijalankan di empat sistem operasi berbeda, seperti Linux, Windows, MacOS, dan juga Solaris.

Untuk mengunduh *software* dari XAMPP dapat dilakukan scanning pada QR Code berikut.



Gambar 3. 12 Situs Download XAMPP

Sumber : <https://www.apachefriends.org/download.html>

Jika dijabarkan, masing-masing huruf yang ada di dalam nama XAMPP menurut para ahli memiliki arti sebagai berikut ini:

1. X (*Cross Platform*)

Merupakan kode penanda untuk *software cross platform* atau yang bisa berjalan di banyak sistem operasi. Jadi, ada XAMPP untuk Windows, MacOS, Linux. Semua itu bersifat

free download XAMPP.

2. A (*Apache*)

Apache adalah aplikasi *web server* yang bersifat gratis dan bisa dikembangkan oleh banyak orang (*open source*).

3. M (*MySQL/MariaDB*)

MySQL atau MariaDB merupakan aplikasi *database server* yang dikembangkan oleh orang yang sama. MySQL berperan dalam mengolah, mengedit, dan menghapus daftar melalui *database*.

4. P (*PHP*)

Huruf “P“ yang pertama dari akronim kata XAMPP adalah inisial untuk menunjukkan eksistensi bahasa pemrograman PHP. Bahasa pemrograman ini biasanya digunakan untuk membuat *website* dinamis, contohnya dalam *website* berbasis CMS WordPress.

5. P (*Perl*)

Sementara itu, huruf P selanjutnya merupakan singkatan dari bahasa pemrograman Perl yang kerap digunakan untuk memenuhi berbagai macam kebutuhan. Perl ini bisa berjalan di dalam banyak digunakan.

Program aplikasi XAMPP berfungsi sebagai *server* lokal untuk mengampu berbagai jenis data *website* yang sedang dalam proses pengembangan. Dalam prakteknya, XAMPP digunakan untuk menguji kinerja fitur ataupun menampilkan konten yang ada di dalam *website* kepada orang lain tanpa harus terkoneksi dengan internet, cukup akses melalui XAMPP *Control panel*, atau istilahnya *website offline*. XAMPP bekerja secara *offline* layaknya *web hosting* biasa namun tidak bisa diakses oleh banyak orang.

3.3.3 Jupyter Notebook



Gambar 3. 13 Logo Jupyter Notebook

Sumber : <https://icon-icons.com/id/icon/jupyter-logo/169453>

Jupyter Notebook adalah lingkungan komputasi interaktif berbasis *web* untuk membuat dokumen *notebook* jupyter. Jupyter notebook mendukung beberapa bahasa seperti Python (ipython), Julia, R, dan lain-lain. Sebagian besar digunakan untuk analisis data, visualisasi data dan komputasi eksplorasi interaktif lebih lanjut.

Untuk mengunduh *software* dari Jupyter Notebook dapat dilakukan scanning pada QR Code berikut.



Gambar 3. 14 Situs Download Jupyter Notebook

Sumber : <https://jupyter.org/install>

Awalnya dikembangkan untuk aplikasi *data science* yang ditulis dengan Python, R, dan Julia, Jupyter Notebook berguna dalam segala macam cara untuk semua jenis proyek :

1. Visualisasi Data
Kebanyakan orang memiliki eksposur pertama mereka ke

Notebook Jupyter melalui visualisasi data, sebuah *notebook* bersama yang mencakup *rendering* dari beberapa data yang ditetapkan sebagai grafik. Jupyter Notebook memungkinkan untuk membuat visualisasi, tetapi juga membagikannya dan memungkinkan perubahan interaktif pada kode dan kumpulan data yang dibagikan.

2. Berbagi Kode

Layanan *cloud* seperti GitHub dan Pastebin menyediakan cara untuk berbagi kode, tetapi sebagian besar non-interaktif. Dengan Jupyter Notebook dapat dilihat kode, menjalankannya, dan menampilkan hasilnya langsung di *browser web*.

3. Interaksi Langsung dengan Kode

Kode Notebook Jupyter tidak statis, itu dapat diedit dan dijalankan kembali secara bertahap, dengan umpan balik yang diberikan langsung di *browser*. Notebook juga dapat menyematkan kontrol pengguna (Misalnya Slider atau bidang input teks) yang dapat digunakan sebagai sumber input untuk kode.

4. Mendokumentasikan Contoh Kode

Jika memiliki sepotong kode dan ingin menjelaskan secara bertahap cara kerjanya, dengan umpan balik langsung di sepanjang jalan, dapat menyematkannya dalam Jupyter Notebook. Kode akan tetap berfungsi penuh, dapat menambahkan interaktivitas bersama dengan penjelasan, yang ditampilkan dan diceritakan pada saat yang sama[16].

3.3.4 Tableau



Gambar 3. 15 Logo Tableau

Sumber : <https://logos-world.net/tableau-logo/>

Tableau adalah *platform business intelligence* untuk membantu

dalam melihat dan memahami data. *Platform* ini merupakan hasil dari proyek ilmu komputer di Stanford yang didirikan pada tahun 2003.

Untuk mengunduh *software* dari Tableau dapat dilakukan scanning pada QR Code berikut.



Gambar 3. 16 Situs Download Tableau

Sumber: <https://www.tableau.com/products/desktop/download>

Tujuan dibuatnya Tableau adalah untuk meningkatkan aliran analisis dan membuat data lebih dapat diakses oleh orang-orang melalui visualisasi. Berikut keunggulan menggunakan Tableau :

1. Visualisasi yang mudah dan interaktif
Keunggulan yang membuat populer Tableau adalah visualisasinya mudah, interaktif, dan menarik. Jika *tools* lain hanya menyajikan grafik biasa, *tools* ini bisa menyajikan grafik bergerak, sehingga tampilan lebih menarik.
2. *User-friendly*
Jika membuat grafik perlu adanya *coding*, Tableau tidak perlu ada *coding*. Tableau diperuntukan bagi siapapun yang bermain dengan data tapi ingin lebih sederhana dalam menyajikannya.
3. Banyak sumber data yang mampu diolah
Banyak data kompleks yang beraturan maupun tidak beraturan, dan juga berbagi sumber, Tableau bisa mengolah dari berbagai sumber. Bahkan bisa menggabungkan dari berbagai *cloud*, *spreadsheet*, dan *big data*.

4. *Dashboard mobile friendly*
Jika tidak sempat membuka laptop, Tableau bisa diakses melalui *smartphone*. Meski lewat smartphone, *user interfacenya* mudah untuk dioperasikan.
5. Terintegrasi dengan bahasa srkip[17].

3.3.5 Anaconda



Gambar 3. 17 Logo Anaconda
Sumber : <https://www.anaconda.com/>

Anaconda merupakan aplikasi gratis yang memiliki banyak package dan tools atau dengan kata lain merupakan *one place solution platform* untuk *data scientist* agar memudahkan dalam mengerjakan proyek karena di dalamnya terdapat berbagai macam *software*. Anaconda bisa dikatakan sebagai paket distribusi python dari *continuum analytics* yang berisi paket *Python* ditambah beberapa paket tambahan untuk keperluan pemrograman *data science*, matematika, hingga teknik dalam satu distribusi yang *user friendly*.

Untuk mengunduh *software* dari Anaconda dapat dilakukan scanning pada QR Code berikut.



Gambar 3. 18 Situs Download Anaconda
Sumber : <https://www.anaconda.com/products/distribution>

3.3.6 Kaggle



Gambar 3. 19 Logo Kaggle

Sumber : https://commons.wikimedia.org/wiki/File:Kaggle_logo.png

Kaggle adalah situs dan platform untuk berlomba membuat model terbaik untuk menganalisa dan memprediksi suatu dataset. Dataset ini diberikan oleh suatu perusahaan, dengan suatu deskripsi masalah tertentu. Misalnya, diberikan data rumah beserta fitur-fiturnya dan harga jualnya, dan deskripsi masalahnya adalah buatlah model untuk memprediksi harga jual berdasarkan fitur-fitur tersebut[18].

Untuk mengunjungi situs resmi Kaggle dapat dilakukan scanning pada QR Code berikut.



Gambar 3. 20 Situs Resmi Kaggle

Sumber : <https://www.kaggle.com/>

3.4 Perangkat Keras Yang Dibutuhkan

Adapun perangkat keras yang diperlukan dalam penunjang pembuatan aplikasi prediksi kanker payudara. Perangkat keras tersebut dijelaskan dengan saran rekomendasi spesifikasi. Berdasarkan hal tersebut, berikut merupakan perangkat keras dan spesifikasinya dalam bentuk tabel.

Tabel 3. 1 Spesifikasi dan Kebutuhan Perangkat Keras

No	Nama Perangkat	Spesifikasi	Keterangan
1	Hardisk	200 GB	Media untuk menyimpan data aplikasi yang dibuat
2	Memory	4 GB	<i>Memory System</i> yang digunakan
3	Processor	AMD A4-9120 RADEON R3, 4 COMPUTE CORES 2C+2G 2.20 GHz	Untuk kecepatan transfer data dari sistem yang sangat bergantung pada kecepatan prosesor komputer
4	Infrastruktur jaringan	PC	Mengolah, menginput serta menghasilkan <i>output</i> data ataupun informasi yang sesuai dengan keinginan pengguna (<i>user</i>)

BAB 4

Perancangan Sistem UML dalam Membuat Aplikasi

4.1 Analisis

Analisis sistem adalah pembagian suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya yang bertujuan untuk mengevaluasi masalah atau kendala yang terjadi pada suatu sistem, sehingga nantinya dapat dilakukan perbaikan atau pengembangan pada sistem tersebut.

Perancangan sistem adalah kegiatan merancang dan mendesain suatu sistem yang baik, yang dimana kegiatan tersebut adalah langkah-langkah operasi dalam proses pengolahan data dan prosedur-prosedur untuk mendukung operasi sistem tersebut.

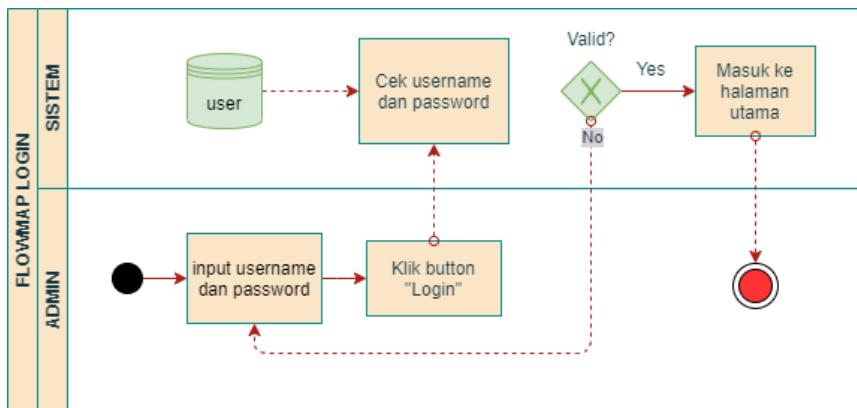
Tujuan dari perancangan sistem adalah untuk memenuhi kebutuhan para pengguna sistem serta memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada programmer dan ahli-ahli yang terlibat didalamnya. Pada bagian ini, dibahas tentang analisis prosedur yang digambarkan dalam bentuk flowmap BPMN, pengkodean, analisis sistem fungsional, dan analisis sistem non fungsional yang meliputi perangkat keras dan perangkat lunak yang digunakan. Selain itu, pada bagian ini juga akan dibahas mengenai analisis pengguna yang terlibat dalam aplikasi tersebut. Tahapan ini sangat penting dalam membantu melanjutkan tahapan selanjutnya yaitu tahapan perancangan.

4.1.1 Analisis Sistem Yang Akan di Bangun

1. *Flowmap Login*

Pada bagian ini, akan dijelaskan tentang aplikasi website yang akan dibangun untuk memprediksi kanker payudara pada pasien berdasarkan faktor-faktor spesifik. Analisis sistem yang sedang berjalan bertujuan untuk memberikan gambaran dan mengetahui lebih lanjut bagaimana cara kerja sistem tersebut. Strategi yang digunakan dalam menganalisis sistem yang akan dibangun ini adalah dengan memecah atau menterjemahkan ke dalam bentuk flowmap BPMN.

Berikut adalah *flowmap login* yang akan dibangun dalam pembuatan aplikasi prediksi kanker payudara.



Gambar 4. 1 Flowmap Login Aplikasi

Keterangan :

1. Admin dapat membuka Aplikasi Prediksi Kanker Payudara
2. Aplikasi menampilkan halaman *login*.
3. Admin dapat melakukan penginputan *username* serta *password*.
4. Setelah itu, sistem dapat melakukan pengecekan apakah *username* serta *password* yang diinputkan admin valid

atau tidak.

5. Jika *username* dan *password* sesuai, maka admin dapat masuk ke halaman utama aplikasi. Sedangkan jika *username* dan *password* tidak sesuai, maka admin akan tetap berada di halaman *login*.

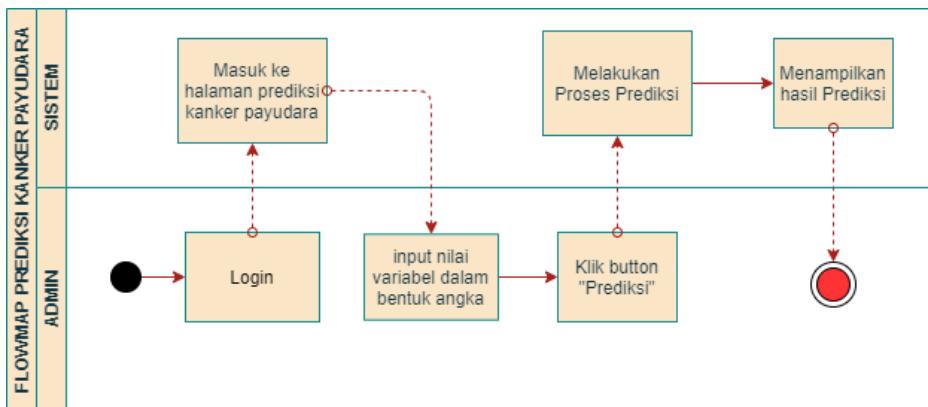
2. **Flowmap Prediksi Kanker Payudara**

Pada bagian ini menjelaskan tentang *flowmap* proses prediksi kanker payudara yang dilakukan oleh admin. *Flowmap* tersebut merupakan gambaran alur proses prediksi kanker payudara oleh admin yang akan dibangun pada aplikasi ini.

Pada *flowmap* ini, terdapat dua pelaku dalam kegiatan, diantaranya adalah:

- a. Admin sebagai pengguna sistem.
- b. Sistem sebagai komponen informasi yang akan digunakan.

Berikut adalah *flowmap* proses prediksi kanker payudara yang akan dibangun dalam pembuatan pada aplikasi.



Gambar 4. 2 Flowmap Prediksi Kanker Payudara

Keterangan :

1. Admin dapat melakukan *login*.
2. Setelah admin berhasil *login*, aplikasi akan menampilkan halaman *dashboard*.
3. Setelah itu, admin dapat memilih menu Prediksi untuk beralih ke halaman prediksi.
4. Pada halaman prediksi, admin dapat menginputkan angka berupa variabel *form* yang disediakan.
5. Setelah diinputkan, aplikasi akan menampilkan hasil prediksi kanker payudara pasien.

4.2 Perancangan Sistem (UML)

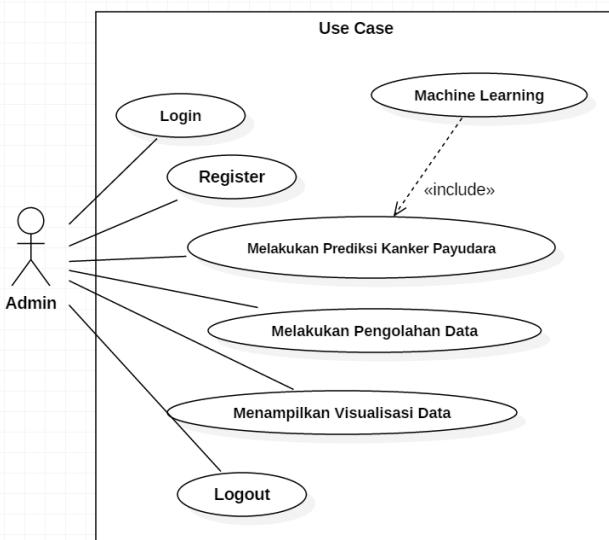
UML adalah sebuah bahasa yang digunakan untuk memodelkan sistem atau aplikasi berbasis objek. Ia merupakan standar spesifikasi yang digunakan untuk dokumentasi, spesifikasi, dan pembangunan sistem perangkat lunak. UML juga merupakan metodologi untuk mengembangkan sistem OOP (Object-Oriented Programming) dan memiliki alat-alat yang membantu dalam pengembangan sistem tersebut. Dengan demikian, UML dapat menjadi cara yang memudahkan dalam pengembangan aplikasi yang berkelanjutan dan bermanfaat sebagai alat bantu untuk mentransfer pengetahuan tentang sistem yang akan dikembangkan dari satu developer ke developer lainnya.

4.2.1 Use case Diagram

Use case diagram adalah salah satu jenis diagram UML (Unified Modeling Language) yang digunakan untuk menggambarkan interaksi sistem dengan aktor-aktor yang terlibat dalam sistem tersebut. Diagram ini memfokuskan pada fungsi sistem dan cara aktor terlibat dengan sistem tersebut.

Use case diagram terdiri dari beberapa elemen utama, yaitu:

1. Aktor: merupakan entitas yang terlibat dalam interaksi dengan sistem, baik itu manusia maupun perangkat lain. Aktor dapat diwakili oleh manusia, organisasi, atau perangkat lain yang menggunakan sistem.
2. Use case: merupakan tindakan atau kegiatan yang dapat dilakukan oleh sistem dengan bantuan aktor. Use case menggambarkan bagaimana sistem memenuhi kebutuhan aktor.
3. Association: merupakan garis yang menghubungkan aktor dengan use case. Association menggambarkan bagaimana aktor terlibat dalam use case.
4. Generalization: merupakan garis yang menghubungkan use case dengan use case lainnya. Generalization menggambarkan bahwa use case yang terhubung memiliki kesamaan atau kemiripan dalam tindakannya.



Gambar 4. 3 Use Case Diagram

A. Definisi Aktor

Pada bagian ini akan dijelaskan aktor-aktor yang terlibat dalam Sistem.

Tabel 4. 1 Definisi Aktor

No	Aktor	Deskripsi
1	Admin	<ul style="list-style-type: none">• <i>Login</i>• Melakukan Prediksi Kanker Payudara• Melakukan Pengolahan Data• Melakukan Visualisasi Data• <i>Logout</i>

B. Definisi *Use case*

Tabel 4. 2 Definisi Use Case

No	Aktor	Deskripsi
1	<i>Machine learning</i>	Merupakan aktivitas interaksi antara manusia dengan mesin. Dalam hal ini, admin membuat model prediksi <i>Machine learning</i> .
2	<i>Login</i>	Merupakan aktivitas <i>login</i> yang dilakukan oleh admin. Sebelum admin masuk ke

		aplikasi ia harus melakukan <i>login</i> terlebih dahulu.
3	Registrasi	Merupakan aktivitas registrasi yang dilakukan oleh admin. Apabila admin belum memiliki akun, maka admin harus melakukan registrasi terlebih dahulu.
4	Melakukan Prediksi Kanker Payudara	Merupakan aktivitas memprediksi kanker pada pasien yang dilakukan oleh admin dengan parameter yang telah diperiksa oleh dokter.
5	Melakukan Pengolahan Data	Merupakan aktivitas <i>insert, read, update</i> dan <i>delete</i> data pasien yang dilakukan oleh admin.
6	Menampilkan Visualisasi Data	Merupakan aktivitas visualisasi grafik data pasien yang dilakukan oleh admin.
7	<i>Logout</i>	Merupakan aktivitas <i>Logout</i> yang dilakukan oleh admin. Admin dapat keluar dari aplikasi jika telah selesai melakukan pekerjaan pada aplikasi.

C. Skenario *Use case*

Skenario *use case* diharapkan setelah berjalannya fungsional *use case*. Selain itu juga diberikan ulasan yang berkaitan dengan tanggapan dari sistem atas suatu aksi yang dilakukan oleh aktor. Setiap *use case* akan diberikan sebuah skenario yang akan menjelaskan secara detail interaksi yang ada di dalamnya.

Tabel 4. 3 Skenario Use case Machine Learning

Identifikasi	
Nomor	1
Nama	<i>Machine learning</i>
Tujuan	Pembuatan model prediksi
Deskripsi	
Aktor	Admin
Skenario Utama	
Kondisi Awal	
Aksi Aktor	Reaksi Sistem
Membuat model prediksi	Melakukan olah data pasien
Kondisi Akhir	Masuk pada aplikasi

Tabel 4. 4 Skenario Use case Login Admin

Identifikasi	
Nomor	1
Nama	<i>Login</i>
Tujuan	Sebelum masuk pada aplikasi
Deskripsi	

Aktor	Admin
Skenario Utama	
Kondisi Awal	
Aksi Aktor	Reaksi Sistem
Input <i>username</i> dan <i>password</i>	Mengecek data dan menentukan aktor tersebut admin atau <i>user</i>
Kondisi Akhir	Masuk pada aplikasi

Tabel 4. 5 Skenario Use case Registrasi

Identifikasi	
Nomor	1
Nama	Registrasi
Tujuan	Membuat akun admin
Deskripsi	
Aktor	Admin
Skenario Utama	
Kondisi Awal	
Aksi Aktor	Reaksi Sistem
Input <i>username</i> , dan <i>password</i>	Menyimpan data admin atau <i>user</i>
Kondisi Akhir	Masuk pada halaman <i>login</i>

Tabel 4. 6 Skenario Use case Melakukan Prediksi

Identifikasi	
Nomor	1
Nama	Melakukan Prediksi Kanker Payudara
Tujuan	Melakukan Prediksi Kanker Payudara
Deskripsi	
Aktor	Admin
Skenario Utama	
Kondisi Awal	
Aksi Aktor	Reaksi Sistem
Menginput Data	Melakukan Prediksi Kanker Payudara
Kondisi Akhir	Admin dapat memprediksi Kanker Payudara

Tabel 4. 7 Skenario Use case Melakukan Visualisasi Data

Identifikasi	
Nomor	1
Nama	Melakukan Visualisasi Data
Tujuan	Melakukan visualisasi data Pasien
Deskripsi	
Aktor	Admin
Skenario Utama	

Kondisi Awal	
Aksi Aktor	Reaksi Sistem
Input data kanker	Menampilkan grafik visualisasi data kanker payudara
Kondisi Akhir	Admin dapat memvisualisasikan data

Tabel 4. 8 Skenario Use case Melakukan Pengolahan Data

Identifikasi	
Nomor	1
Nama	Melakukan Pengolahan Data
Tujuan	Melakukan pengolahan data Pasien
Deskripsi	
Aktor	Admin
Skenario Utama	
Kondisi Awal	
Aksi Aktor	Reaksi Sistem
<i>Insert, read, update, dan delete</i> data kanker.	Mengubah data Pasien yang diubah oleh admin
Kondisi Akhir	Admin dapat mengelola data

Tabel 4. 9 Skenario Use case Logout Admin

Identifikasi	
Nomor	1
Nama	<i>Logout</i>
Tujuan	Keluar dari aplikasi
Deskripsi	
Aktor	Admin
Skenario Utama	
Kondisi Awal	
Aksi Aktor	Reaksi Sistem
Menekan tombol <i>Logout</i>	Melakukan proses keluar dari aplikasi
Kondisi Akhir	Keluar dari aplikasi

4.2.2 Class Diagram

Class diagram adalah salah satu jenis diagram UML (Unified Modeling Language) yang digunakan untuk menggambarkan struktur dan relasi antar kelas dalam suatu sistem. Diagram ini menggambarkan bagaimana kelas-kelas tersebut saling berinteraksi dan berkaitan satu sama lain dalam sistem yang dibuat.

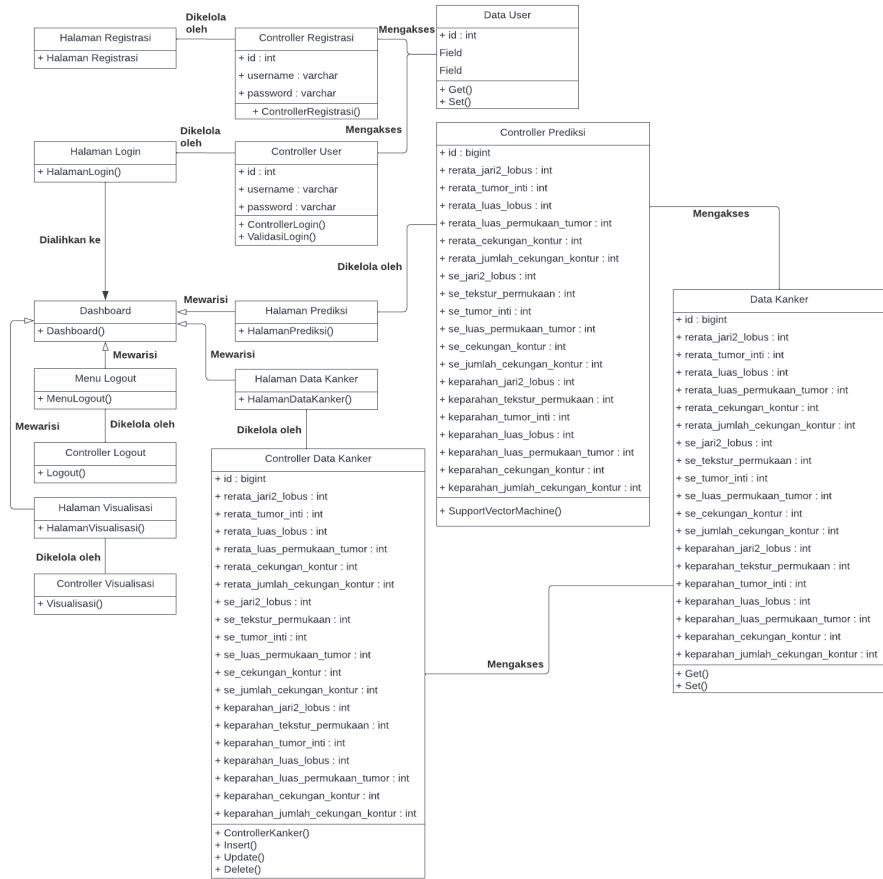
Class diagram terdiri dari beberapa elemen utama, yaitu:

1. Class: merupakan entitas yang mewakili suatu objek dalam sistem. Class menggambarkan atribut (properti) dan operasi (method) yang dimiliki oleh objek tersebut.
2. Attribute: merupakan properti yang dimiliki oleh sebuah

kelas. Attribute menggambarkan data yang disimpan oleh kelas tersebut.

3. Operation: merupakan method yang dimiliki oleh sebuah kelas. Operation menggambarkan tindakan atau kegiatan yang dapat dilakukan oleh kelas tersebut.
4. Association: merupakan garis yang menghubungkan satu kelas dengan kelas lainnya. Association menggambarkan relasi atau hubungan antar kelas.
5. Generalization: merupakan garis yang menghubungkan kelas dengan superclass-nya. Generalization menggambarkan bahwa kelas tersebut merupakan turunan dari superclass yang terhubung.

Berikut adalah *class diagram* dalam bentuk gambar.



Gambar 4. 4 Class Diagram

4.2.3 Activity Diagram

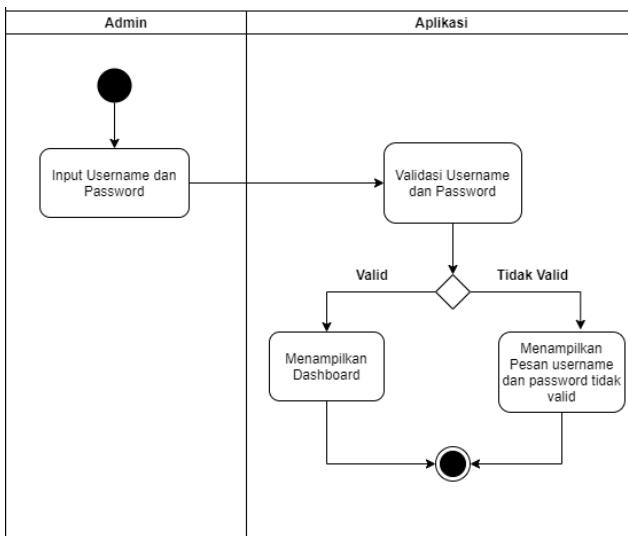
Activity diagram adalah salah satu jenis diagram UML (Unified Modeling Language) yang digunakan untuk menggambarkan aliran aktivitas dalam suatu sistem. Diagram ini menggambarkan langkah-langkah yang dilakukan dalam suatu proses atau kegiatan, serta kondisi yang dapat mempengaruhi aliran aktivitas tersebut.

A. *Activity diagram Login Admin*

Berikut ini merupakan *Activity diagram Login Admin* yang menggambarkan tentang berbagai aktifitas yang terjadi pada suatu sistem. Dimana pada diagram ini, terdiri dari komponen-komponen sebagai berikut.

1. Aktor : Admin dan Aplikasi
2. Satu *initial node*
3. *Activity* : Terdapat empat *activity* yang dapat dipaparkan sebagai berikut.
 - a. Input *username* dan *password*
 - b. Validasi *username* dan *password*
 - c. Menampilkan *dashboard*
 - d. Menampilkan pesan *username* dan *password* tidak valid
4. Satu percabangan untuk valid dan tidak valid
5. Satu *Activity Final Node*

Berikut adalah *activity diagram* melakukan *login* admin dalam bentuk gambar.



Gambar 4. 5 Activity diagram Login Admin

Keterangan :

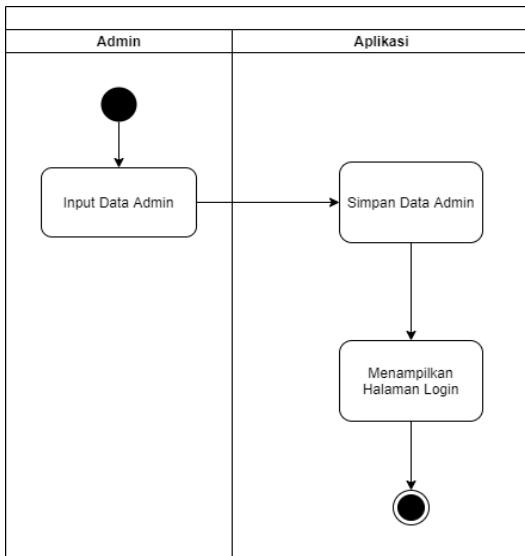
1. Admin menginput *username* dan *password*.
2. Aplikasi melakukan validasi *username* dan *password*.
3. Jika valid akan menampilkan halaman *dashboard*.
4. Jika tidak valid menampilkan pesan *username* dan *password* tidak valid.

B. *Activity diagram* Melakukan Registrasi Admin

Berikut ini merupakan *Activity diagram* Registrasi Admin yang menggambarkan tentang berbagai aktifitas yang terjadi pada suatu sistem. Dimana pada diagram ini, terdiri dari komponen-komponen sebagai berikut.

1. Aktor : Admin dan Aplikasi
2. *Satu initial node*

3. *Activity* : Terdapat tiga *activity* yang dapat dipaparkan sebagai berikut.
 - a. Input data admin
 - b. Simpan data admin
 - c. Menampilkan halaman *login*
4. Satu *Activity Final Node*



Gambar 4. 6 Activity diagram Melakukan Registrasi Admin

Keterangan :

1. Admin menginput data admin.
2. Aplikasi menyimpan data admin.
3. Aplikasi menampilkan halaman *login*.

C. *Activity diagram* Melakukan Pengolahan Data

Berikut ini merupakan *Activity diagram* Pengolahan Data yang menggambarkan tentang berbagai aktifitas yang terjadi pada suatu sistem. Dimana pada diagram ini, terdiri dari komponen-komponen sebagai berikut.

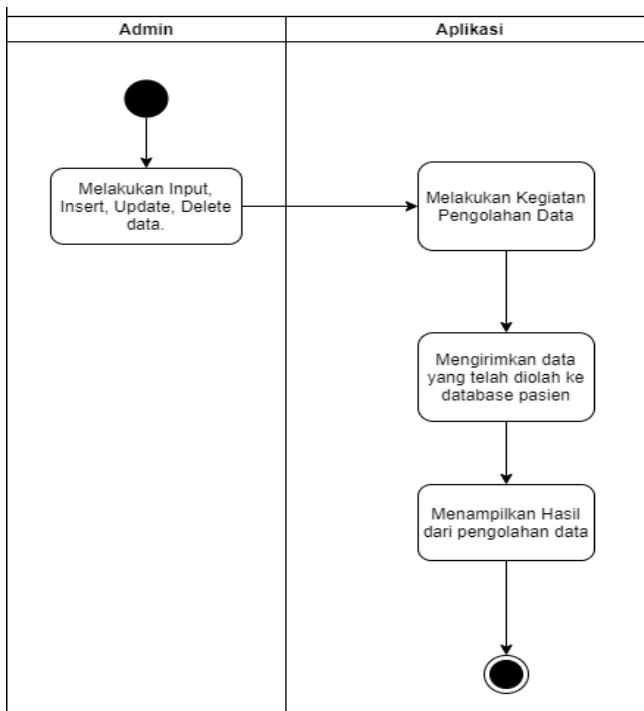
1. Aktor : Admin dan Aplikasi
2. Satu *initial node*
3. *Activity* : Terdapat empat *activity* yang dapat dipaparkan sebagai berikut.
 - a. Melakukan *insert*, *update*, atau *delete* data
 - b. Melakukan proses olah data yang dilakukan
 - c. Mengirim data yang diolah ke *database* pegawai
 - d. Menampilkan hasil pengolahan data
4. Satu *Activity Final Node*

Berikut adalah *activity diagram* melakukan pengolahan data dalam bentuk gambar.

Keterangan :

1. Admin melakukan olah data seperti *insert*, *update*, dan *delete* data.
2. Aplikasi melakukan proses olah data yang dilakukan.
3. Aplikasi mengirim data yang diolah ke *database* pegawai.

Aplikasi menampilkan data hasil *create/update/delete*.



Gambar 4. 7 Activity diagram Melakukan Pengolahan Data

D. *Activity diagram* Melakukan Prediksi Kanker Payudara

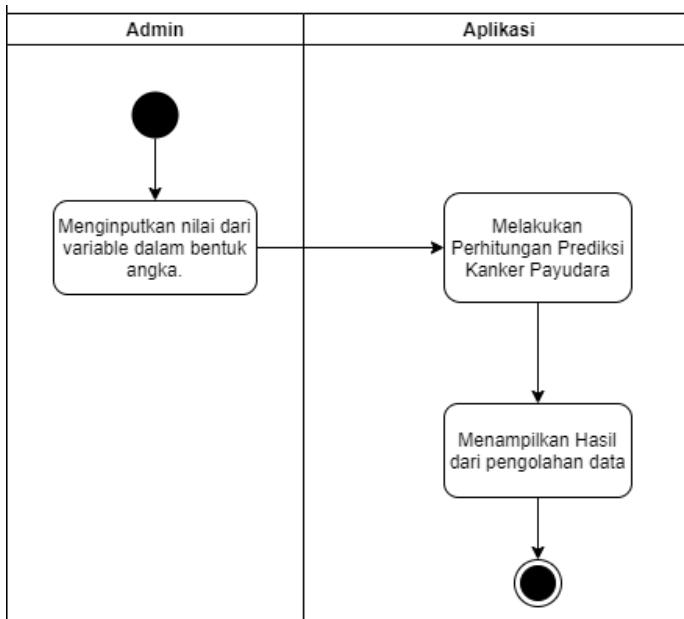
Berikut ini merupakan *Activity diagram* Melakukan Prediksi Kanker yang menggambarkan tentang berbagai aktifitas yang terjadi pada suatu sistem. Dimana pada diagram ini, terdiri dari komponen-komponen sebagai berikut.

1. Aktor : Admin dan Aplikasi
2. Satu *initial node*
3. *Activity* : Terdapat tiga *activity* yang dapat dipaparkan sebagai berikut.

- Menginput variabel independen dalam bentuk angka
- Melakukan perhitungan prediksi kanker
- Menampilkan hasil prediksi kanker

4. Satu Activity Final Node

Berikut adalah *activity diagram* melakukan prediksi kanker payudara pada pasien dalam bentuk gambar.



Gambar 4. 8 Activity diagram Melakukan Prediksi Kanker Payuda

Keterangan :

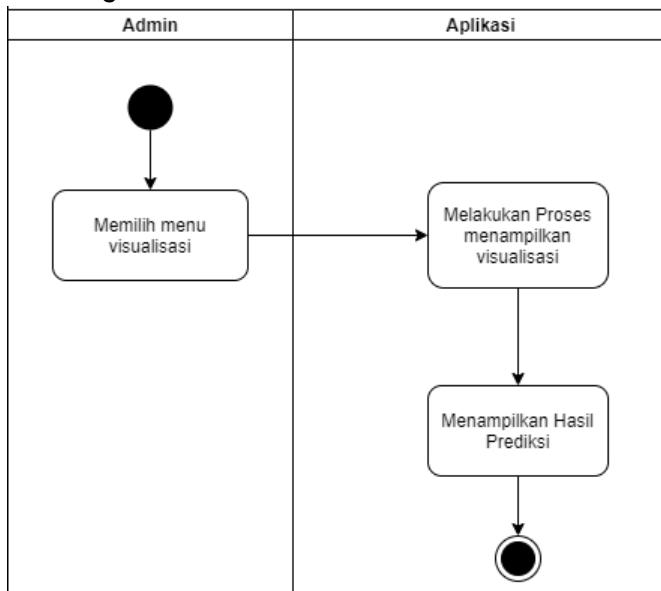
1. Admin menginputkan nilai dari variabel dalam bentuk angka pada *form* prediksi.
2. Aplikasi melakukan proses perhitungan prediksi kanker.
3. Aplikasi menampilkan hasil prediksi.

E. *Activity diagram* Melakukan Visualisasi

Berikut ini merupakan *Activity diagram* Visualisasi Data yang menggambarkan tentang berbagai aktifitas yang terjadi pada suatu sistem. Dimana pada diagram ini, terdiri dari komponen-komponen sebagai berikut.

1. Aktor : Admin dan Aplikasi
2. Satu *initial node*
3. *Activity* : Terdapat tiga *activity* yang dapat dipaparkan sebagai berikut.
 - a. Memilih menu visualisasi
 - b. Melakukan proses menampilkan visualisasi
 - c. Menampilkan halaman visualisasi
4. Satu *Activity Final Node*

Berikut adalah *activity diagram* melakukan pengolahan data dalam bentuk gambar



Gambar 4. 9 Activity diagram Melakukan Visualisasi

Keterangan :

1. Admin memilih menu visualisasi.
2. Aplikasi melakukan proses menampilkan visualisasi.
3. Aplikasi menampilkan halaman visualisasi.

F. *Activity diagram Logout Admin*

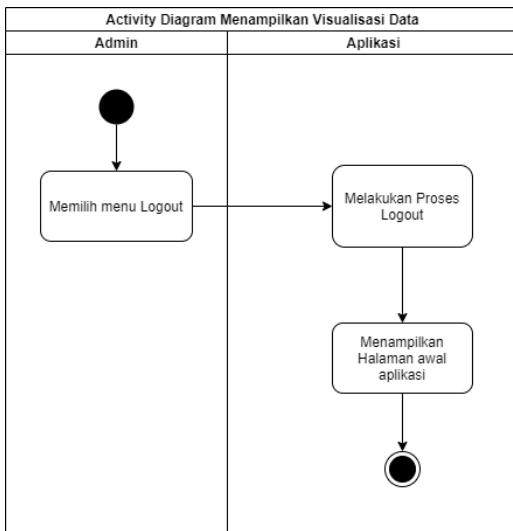
Berikut ini merupakan *Activity diagram Logout Admin* yang menggambarkan tentang berbagai aktifitas yang terjadi pada suatu sistem. Dimana pada diagram ini, terdiri dari komponen-komponen sebagai berikut.

1. Aktor : Admin dan Aplikasi
2. Satu *initial node*
3. *Activity* : Terdapat tiga *activity* yang dapat dipaparkan sebagai berikut.
 - a. Memilih menu *Logout*
 - b. Melakukan proses *Logout* aplikasi
 - c. Menampilkan halaman awal aplikasi
4. Satu *Activity Final Node*

Keterangan :

1. Admin memilih menu *Logout*.
2. Aplikasi melakukan proses *Logout* aplikasi.
3. Aplikasi menampilkan halaman awal aplikasi.

Berikut adalah *activity diagram* melakukan pengolahan data dalam bentuk gambar.



Gambar 4. 10 Activity diagram Logout Admin

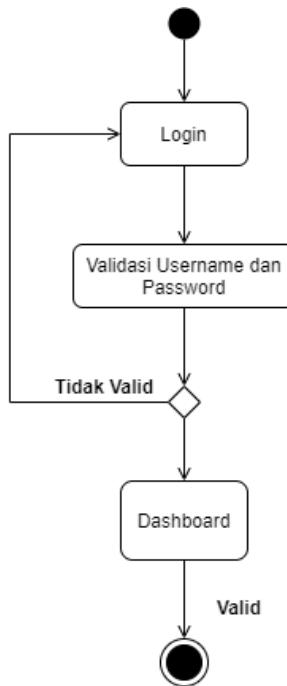
4.2.4 Statechart Diagram

Diagram statechart dapat digunakan untuk menggambarkan bagaimana suatu objek dalam sistem dapat berpindah dari satu keadaan ke keadaan lainnya sebagai akibat dari stimulus yang diterimanya. Biasanya, diagram statechart dapat digunakan untuk menggambarkan perilaku dari suatu kelas tertentu (satu kelas dapat memiliki lebih dari satu diagram statechart)." Jika ada sesuatu yang tidak jelas, silakan tanyakan kembali. Saya akan senang untuk membantu Anda.

A. Statechart Diagram Login Admin

Berikut ini merupakan *Activity diagram Login Admin* yang menggambarkan transisi dan perubahan keadaan dari suatu objek pada sistem.

Berikut adalah *Statechart Diagram* melakukan *login* admin dalam bentuk gambar.



Gambar 4. 11 Statechart Diagram melakukan login admin

Keterangan :

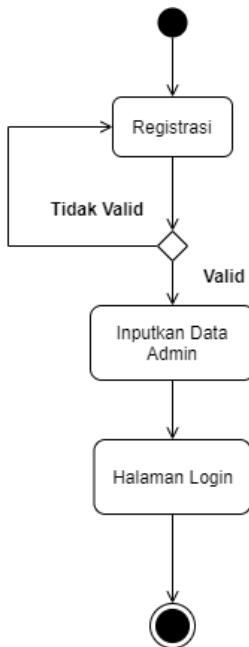
1. Admin melakukan *login*.
2. Aplikasi melakukan validasi *username* dan *password*.
3. Jika valid akan menampilkan halaman *dashboard*.
4. Jika tidak valid akan dikembalikan ke halaman *login*.

B. *Statechart Diagram* Registrasi Admin

Berikut ini merupakan *Activity diagram* Registrasi Admin

yang menggambarkan transisi dan perubahan keadaan dari suatu objek pada sistem.

Berikut adalah *Statechart Diagram* melakukan registrasi admin dalam bentuk gambar.



Gambar 4. 12 Statechart Diagram melakukan registrasi admin

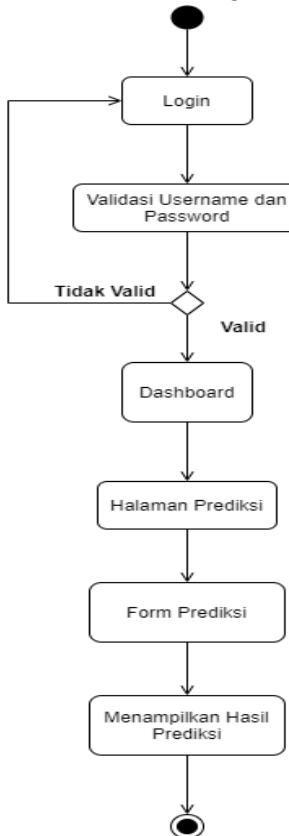
Keterangan :

1. Admin menginput data admin.
2. Jika valid maka data admin akan terinput ke *database*.
3. Jika tidak valid maka akan dikembalikan ke halaman registrasi admin.
4. Aplikasi menampilkan halaman *login*.

C. *Statechart Diagram* Melakukan Prediksi

Berikut ini merupakan *Activity diagram* Prediksi Kanker Payudara yang menggambarkan transisi dan perubahan keadaan dari suatu objek pada sistem.

Berikut adalah *Statechart Diagram* melakukan prediksi kanker payudara dalam bentuk gambar.



Gambar 4. 13 Statechart Diagram melakukan prediksi kanker payudara

Keterangan :

1. Admin melakukan *login*.
2. Aplikasi melakukan validasi *username* dan *password*.
3. Jika valid akan menampilkan halaman *dashboard*.
4. Jika tidak valid akan dikembalikan ke halaman *login*.
5. Aplikasi menampilkan *form* prediksi pada halaman prediksi
6. Aplikasi menampilkan hasil prediksi kanker payudara.

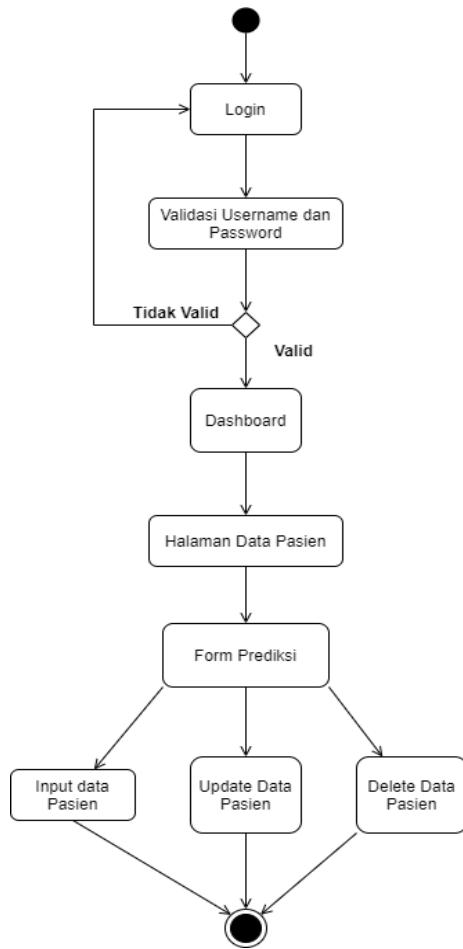
D. *Statechart Diagram* Pengolahan Data

Berikut ini merupakan *Activity diagram* Pengolahan Data Pasien yang menggambarkan transisi dan perubahan keadaan dari suatu objek pada sistem.

Berikut adalah *Statechart Diagram* melakukan pengolahan data dalam bentuk gambar.

Keterangan :

1. Admin melakukan *login*.
2. Aplikasi melakukan validasi *username* dan *password*.
3. Jika valid akan menampilkan halaman *dashboard*.
4. Jika tidak valid akan dikembalikan ke halaman *login*.
5. Aplikasi akan menampilkan halaman data kanker.
6. Pada halaman data kanker, admin dapat memilih pengolahan antara input data kanker, *update* data kanker, ataupun *delete* data.



Gambar 4. 14 Statechart Diagram melakukan pengolahan data

Keterangan :

1. Admin melakukan *login*.
2. Aplikasi melakukan validasi *username* dan *password*.
3. Jika valid akan menampilkan halaman *dashboard*.
4. Jika tidak valid akan dikembalikan ke halaman *login*.
5. Aplikasi akan menampilkan halaman data pasien.

6. Pada halaman data pasien, admin dapat memilih pengolahan antara input data kanker, *update* data kanker, ataupun *delete* data kanker.

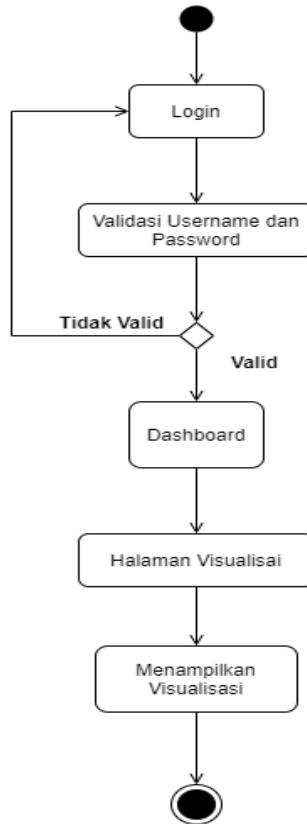
E. *Statechart Diagram* Visualisasi

Berikut ini merupakan *Activity diagram* Visualisasi Data yang menggambarkan transisi dan perubahan keadaan dari suatu objek pada sistem.

Keterangan :

1. Admin melakukan *login*.
2. Aplikasi melakukan validasi *username* dan *password*.
3. Jika valid akan menampilkan halaman *dashboard*.
4. Jika tidak valid akan dikembalikan ke halaman *login*.
5. Aplikasi menampilkan halaman visualisasi.
6. Visualisasi data ditampilkan oleh aplikasi

Berikut adalah *Statechart Diagram* melakukan visualisasi data dalam bentuk gambar.

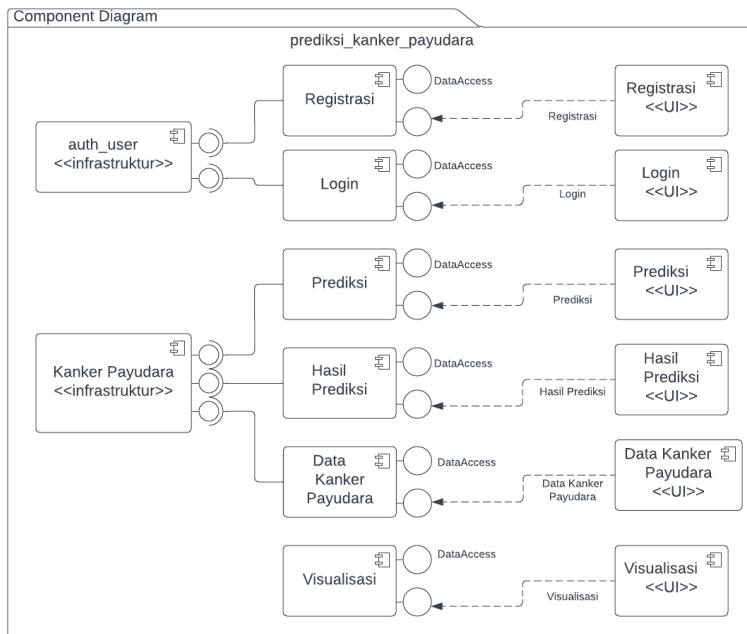


Gambar 4. 15 Statechart Diagram melakukan visualisasi data

4.2.5 Component Diagram

Diagram komponen dapat menggambarkan struktur dan hubungan antara komponen perangkat lunak, termasuk dependensi di antara mereka. Komponen perangkat lunak atau yang biasa disebut dengan perangkat lunak adalah modul yang berisi kode, baik berisi kode sumber maupun kode biner, baik itu library maupun executable, baik yang muncul pada saat compile time, link time, maupun runtime. Pada umumnya, komponen dapat terdiri dari beberapa kelas dan atau paket, tapi juga dapat dibentuk dari komponen-komponen yang lebih kecil.

Berikut adalah *Component Diagram* aplikasi prediksi kanker payudara dalam bentuk gambar.



Gambar 4. 16 Component Diagram

4.2.6 ERD (Entity Relationship Diagram)

ERD (Entity Relationship Diagram) adalah model teknik pendekatan yang menyatakan atau menggambarkan hubungan suatu model[19]. Didalam hubungan ini tersebut dinyatakan yang utama dari ERD adalah menunjukan objek data (Entity) dan hubungan (Relationship), yang ada pada Entity berikutnya[19].

Simbol-simbol dalam ERD (Entity Relationship Diagram) adalah sebagai berikut[19]:

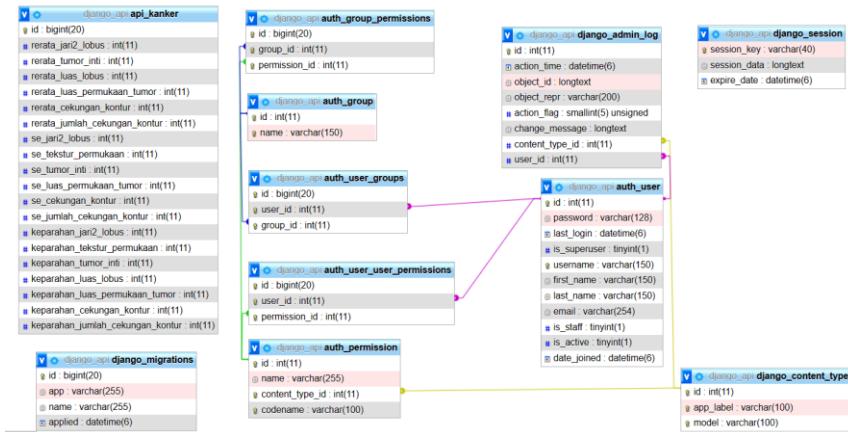
1. Entitas: suatu yang nyata atau abstrak yang mempunyai karakteristik dimana kita akan menyimpan data.
2. Atribut: ciri umum semua atau sebagian besar instansi pada entitas tertentu.
3. Relasi: hubungan alamiah yang terjadi antara satu atau lebih entitas.
4. Link: garis penghubung atribut dengan kumpulan entitas dan kumpulan entitas dengan relasi.

Pada ERD ini terdapat 11 (sebelas) entitas (tabel), entitas tersebut dapat dijelaskan sebagai berikut.

1. Entitas berelasi
 - a. Auth_user_groups
 - b. Auth_group
 - c. Auth_user
 - d. Auth_user_user_permissions
 - e. Auth_group_permissions
 - f. Django_admin_log
 - g. Django_content_type
 - h. Auth_permission
2. Entitas tidak berelasi
 - a. Django_session
 - b. Django_migrations

c. Api_kanker

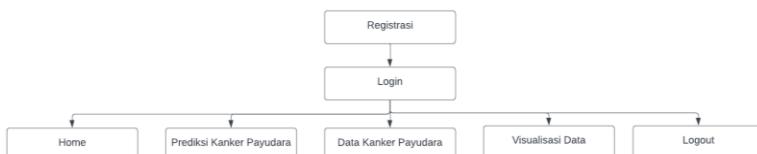
Berikut adalah *Entity Relationship Diagram* aplikasi prediksi kanker payudara dalam bentuk gambar.



Gambar 4. 17 Entity Relationship Diagram aplikasi prediksi kanker payudara

4.3 Struktur Menu

Struktur menu ialah bentuk umum dari suatu rancangan aplikasi dalam memudahkan pengguna untuk menjalankan aplikasi. Sehingga saat menjalankan aplikasi, pengguna tidak mengalami kesulitan dalam memilih menumenu yang diinginkan. Berikut adalah struktur menu dari aplikasi Prediksi Kanker Payudara secara keseluruhan.



Gambar 4. 18 Struktur Menu

Berdasarkan gambar diatas, terdapat dijelaskan aplikasi memiliki halaman *dashboard* yang dapat mengkases menu lainnya yaitu sebagai berikut.

1. Halaman Registrasi

Halaman Registrasi adalah halaman yang digunakan untuk mendaftarkan diri untuk mengakses aplikasi.

2. Halaman Login

Halaman Login adalah halaman yang digunakan untuk validasi pengguna apakah valid atau tidak.

3. Halaman Home

Home adalah halaman awal pada aplikasi.

4. Halaman Prediksi Kanker

Halaman Prediksi Kanker adalah halaman untuk melakukan prediksi kanker payudara.

5. Halaman Data Kanker

Halaman Data Kanker adalah halaman yang menampilkan kumpulan data kanker.

6. Halaman Visualisasi Data

Halaman Visualisasi Data adalah halaman yang menampilkan visualisasi data.

7. *Logout*.

Logout adalah menu yang digunakan oleh *user* untuk keluar aplikasi.

BAB 5

Implementasi Aplikasi

5.1 Implementasi Kode Aplikasi

Pada bagian implementasi kode aplikasi, dipaparkan kode-kode pada setiap proses berjalannya sistem yang telah dibangun. Pada bagian ini, pemaparan pola desain arsitektur dalam sistem. Arsitektur tersebut ialah *Views* dan *Template* serta URL.

5.1.1 Implementasi Package Yang Digunakan

```
from django.db import migrations, models
from django.apps import AppConfig
from apk_predict.models import Kanker
from django import forms

from django.db import models
from django.shortcuts import render, redirect
import pandas as pd

import pickle

from django.views import View

from .models import Kanker

from django.http.response import JsonResponse
from django.utils.decorators import method_decorator
from django.views.decorators.csrf import csrf_exempt
import json

from apk_predict.forms import KankerForm
from apk_predict.models import Kanker
from pathlib import Path
import os

from django.contrib import admin
from django.urls import path
from apk_predict import views
from django.conf import settings
```

Kode diatas adalah kode dari package yang digunakan dalam perancangan aplikasi. Package yang digunakan merupakan sesuai kebutuhan dalam perancangan aplikasi.

5.1.2 Implementasi Proses Login

A. Views

```
32     def LoginPage(request):
33         if request.method=='POST':
34             username=request.POST.get('username')
35             pass1=request.POST.get('pass')
36             User=authenticate(request,username=username,password=pass1)
37             if User is not None:
38                 login(request,User)
39                 return redirect('home')
40             else:
41                 return HttpResponse("Username or Password is incorrect!!!"
42
43     return render (request,'login.html')
44
```

B. Template

```
registration / templates / login.html
141     <div class="shape"></div>
142     </div>
143     <form method="post">
144         <h3>Login Here</h3><br>
145         <br>
146         <center>
147             <a href="{% provider_login_url 'google' %}?next="/">Login With Google</a>
148         </center>
149         {% csrf_token %}
150         <label for="username">Username</label>
151         <input type="text" placeholder="Enter Username" id="username" name="username">
152
153         <label for="password">Password</label>
154         <input type="password" placeholder="Password" id="password" name="pass">
155
156         <button type="submit">Log In</button>
157         {% comment %} <button class="btn btn-outline-info">
158             <a href="{% provider_login_url 'google' %}?next="/">
159                 <i class= "fab fa-google"></i>
160                 Login With Google
161             </a>
162         </button> {% endcomment %}
163         <center>
164             <a href="{% url 'signup' %}">Create a account</a>
165         </center>
166     </form>
167 </body>
168 </html>
```

C. URL

```
24     path('login/',views.LoginPage,name='login'),
```

5.1.3 Implementasi Proses Registrasi

A. Views

```
11  def SignupPage(request):
12      if request.method=='POST':
13          uname=request.POST.get('username')
14          email=request.POST.get('email')
15          pass1=request.POST.get('password1')
16          pass2=request.POST.get('password2')
17
18          if pass1!=pass2:
19              return HttpResponse("You password and confrom password are not Same")
20
21          if User is not None:
22              return HttpResponse("Harap isi dengan benar !!")
23          else:
24              my_user=User.objects.create_user(uname,email,pass1)
25              my_user.save()
26              return redirect('login')
27
```

B. Template

```
registration > templates > signup.html
137 <body>
138     <div class="background">
139         <div class="shape"></div>
140         <div class="shape"></div>
141     </div>
142     <form action="" method="post">
143         {% csrf_token %}
144         <h3>Signup Here</h3>
145
146         <label for="username">Username</label>
147         <input type="text" placeholder="Username" name="username" id="username">
148
149         <label for="email">Email</label>
150         <input type="email" placeholder="Email or Phone" name="email" id="email">
151
152         <label for="password1">Password</label>
153         <input type="password" placeholder="Password" id="password1" name="password1">
154
155
156         <label for="password2">Confirm Password</label>
157         <input type="password" placeholder="Confrom Password" id="password2" name="password2">
158         <button type="submit">Signup</button>
159         <center>
160             <a href="{% url 'login' %}">i have already account ! </a>
161         </center>
162     </form>
163 </body>
164 </html>
```

C. URL

```
23     path('' ,views.SignupPage,name='signup'),
```

5.1.4 Implementasi Proses Prediksi Kanker Payudara

A. Views

```
def index(request):
    context = {'a' : 1}
    return render(request, 'index.html', context)
```

B. Template

```
<h3>Prediksi Kanker Payudara</h3>
{% block content %}
<form method="POST" action="result">
    {% csrf_token %}
    <p>Masukan rerata_jari2_lobus :</p>
    <input type="number" name="rerata_jari2_lobus" id="rerata_jari2_lobus" />
    <br />
    <p>Masukan rerata_tumor_inti :</p>
    <input type="number" name="rerata_tumor_inti" id="rerata_tumor_inti" />
    <br />
    <p>Masukan rerata_luas_lobus :</p>
    <input type="number" name="rerata_luas_lobus" id="rerata_luas_lobus" />
    <br />
    <p>Masukan rerata_luas_permukaan_tumor :</p>
    <input
        type="number"
        name="rerata_luas_permukaan_tumor"
        id="rerata_luas_permukaan_tumor"
    />
    <br />
    <p>Masukan rerata_cekungan_kontur :</p>
    <input
        type="number"
        name="rerata_cekungan_kontur"
        id="rerata_cekungan_kontur"
    />
    <br />
    <p>Masukan rerata_jumlah_cekungan_kontur :</p>
    <input
        type="number"
```

```

        name="rerata_jumlah_cekungan_kontur"
        id="rerata_jumlah_cekungan_kontur"
    />
<br />
<p>Masukan se_jari2_lobus :</p>
<input type="number" name="se_jari2_lobus" id="se_jari2_lobus" />
<br />
<p>Masukan se_tekstur_permukaan :</p>
<input
    type="number"
    name="se_tekstur_permukaan"
    id="se_tekstur_permukaan"
/>
<br />
<p>Masukan se_tumor_inti :</p>
<input type="number" name="se_tumor_inti" id="se_tumor_inti" />
<br />
<p>Masukan se_luas_permukaan_tumor :</p>
<input
    type="number"
    name="se_luas_permukaan_tumor"
    id="se_luas_permukaan_tumor"
/>
<br />
<p>Masukan se_cekungan_kontur :</p>
<input type="number" name="se_cekungan_kontur" id="se_cekungan_kontur" />
<br />
<p>Masukan se_jumlah_cekungan_kontur :</p>
<input
    type="number"
    name="se_jumlah_cekungan_kontur"
    id="se_jumlah_cekungan_kontur"
/>
<br />
<p>Masukan keparahan_luas_lobus :</p>
<input
    type="number"
    name="keparahan_luas_lobus"
    id="keparahan_luas_lobus"
/>
<br />
<p>Masukan keparahan_luas_permukaan_tumor :</p>
<input
    type="number"
    name="keparahan_luas_permukaan_tumor"
    id="keparahan_luas_permukaan_tumor"
/>
<br />
<p>Masukan keparahan_cekungan_kontur :</p>
<input
    type="number"
    name="keparahan_cekungan_kontur"
    id="keparahan_cekungan_kontur"
/>
<br />
<p>Masukan keparahan_jumlah_cekungan_kontur :</p>
<input
    type="number"
    name="keparahan_jumlah_cekungan_kontur"
    id="keparahan_jumlah_cekungan_kontur"
/>
<br />
<input type="submit" value="Lakukan Prediksi" />
{%- endblock %}
</form>

```

C. URL

```
[path(' ', views.index, name='homepage'),
```

5.1.5 Implementasi Hasil Prediksi Kanker Payudara

A. Views

```
def result(request):
    print(request)
    rerata_jariz_lobus = int(request.POST.get('rerata_jariz_lobus'))
    rerata_tumor_inti = int(request.POST.get('rerata_tumor_inti'))
    rerata_luas_lobus = int(request.POST.get('rerata_luas_lobus'))
    rerata_luas_permukaan_tumor = int(request.POST.get('rerata_luas_permukaan_tumor'))
    rerata_cekungan_kontur = int(request.POST.get('rerata_cekungan_kontur'))
    rerata_jumlah_cekungan_kontur = int(request.POST.get('rerata_jumlah_cekungan_kontur'))
    se_jariz_lobus = int(request.POST.get('se_jariz_lobus'))
    se_tekstur_permukaan = int(request.POST.get('se_tekstur_permukaan'))
    se_tumor_inti = int(request.POST.get('se_tumor_inti'))
    se_luas_permukaan_tumor = int(request.POST.get('se_luas_permukaan_tumor'))
    se_cekungan_kontur = int(request.POST.get('se_cekungan_kontur'))
    se_jumlah_cekungan_kontur = int(request.POST.get('se_jumlah_cekungan_kontur'))
    se_fraktal_spesimen = int(request.POST.get('se_fraktal_spesimen'))
    keparahan_jariz_lobus = int(request.POST.get('keparahan_jariz_lobus'))
    keparahan_tekstur_permukaan = int(request.POST.get('keparahan_tekstur_permukaan'))
    keparahan_tumor_inti = int(request.POST.get('keparahan_tumor_inti'))
    keparahan_luas_lobus = int(request.POST.get('keparahan_luas_lobus'))
    keparahan_luas_permukaan_tumor = int(request.POST.get('keparahan_luas_permukaan_tumor'))
    keparahan_cekungan_kontur = int(request.POST.get('keparahan_cekungan_kontur'))
    keparahan_jumlah_cekungan_kontur = int(request.POST.get('keparahan_jumlah_cekungan_kontur'))
    model = pd.read_pickle("./models/model.pickle")
    result = model.predict([[rerata_jariz_lobus, rerata_tumor_inti, rerata_luas_lobus, rerata_luas_permukaan_tumor,
    rerata_cekungan_kontur, rerata_jumlah_cekungan_kontur, se_jariz_lobus, se_tekstur_permukaan, se_tumor_inti,
    se_luas_permukaan_tumor, se_cekungan_kontur, se_jumlah_cekungan_kontur, se_fraktal_spesimen, keparahan_jariz_lobus,
    keparahan_tekstur_permukaan, keparahan_tumor_inti, keparahan_luas_lobus, keparahan_luas_permukaan_tumor,
    keparahan_cekungan_kontur, keparahan_jumlah_cekungan_kontur]])
    return render(request, 'result.html', {'result':result})
```

B. Template

```
<h1> Hasil Prediksi</h1>
{{ result }}
```

C. URL

```
path('result', views.result, name='result'),
```

5.1.6 Implementasi Proses CRUD Data Kanker Payudara

5.1.6.1 Halaman Menampilkan Data

A. Views

```
def view(request):
    kanker = Kanker.objects.all()
    return render(request, "view.html", {'kanker': kanker})
```

B. Template

```
<h3>
  <span class="font-weight-light my-1 text-center"
        style="color: #ffff00">
    >
  List Cancer Datas
  </span>
  <a href="http://127.0.0.1:8000/knkr" class="btn btn-dark btn-sm" style="float:right">[+] Tambah Data Kanker</a>
</div>
</div>

<div class="card-body">
  <table class="table table-dark table-striped">
    <thead class="thead-dark">
      <tr>
        <th scope="col">Id</th>
        <th scope="col">Renata Jari2 Lobus</th>
        <th scope="col">Renata Tumor Inti</th>
        <th scope="col">Renata Luas Lobus</th>
        <th scope="col">Renata Luas Permukaan tumor</th>
        <th scope="col">Renata Cekungan Kontur</th>
        <th scope="col">Renata Jumlah Cekungan Kontur</th>
        <th scope="col">SE Jari2 Lobus</th>
        <th scope="col">SE Tekstur Permukaan</th>
        <th scope="col">SE Tumor Inti</th>
        <th scope="col">SE Luas Permukaan</th>
        <th scope="col">SE Cekungan Kontur</th>
        <th scope="col">SE Jumlah Cekungan Kontur</th>
        <th scope="col">SE Fraktal Spesimen</th>
        <th scope="col">Keparahan Jari2 Lobus</th>
        <th scope="col">Keparahan Tekstur Permukaan</th>
      </tr>
      <tr>
        <th scope="col">Keparahan Tumor Inti</th>
        <th scope="col">Keparahan Luas Lobus</th>
        <th scope="col">Keparahan Luas Permukaan Tumor</th>
        <th scope="col">Keparahan Cekungan Kontur</th>
        <th scope="col">Keparahan Jumlah Cekungan Kontur</th>
        <th scope="col">Action</th>
      </tr>
    </thead>
    <% for kanker in kanker %>
      <tr>
        <th scope="row">{{ kanker.id }}</th>
        <td>{{ kanker.renata_jari2_lobus }}</td>
        <td>{{ kanker.renata_tumor_inti }}</td>
        <td>{{ kanker.renata_luas_lobus }}</td>
        <td>{{ kanker.renata_luas_permukaan_tumor }}</td>
        <td>{{ kanker.renata_cekungan_kontur }}</td>
        <td>{{ kanker.renata_jumlah_cekungan_kontur }}</td>
        <td>{{ kanker.se_jari2_lobus }}</td>
        <td>{{ kanker.se_tekstur_permukaan }}</td>
        <td>{{ kanker.se_tumor_inti }}</td>
        <td>{{ kanker.se_luas_permukaan_tumor }}</td>
        <td>{{ kanker.se_cekungan_kontur }}</td>
        <td>{{ kanker.se_jumlah_cekungan_kontur }}</td>
        <td>{{ kanker.se_fraktal_spesimen }}</td>
        <td>{{ kanker.keparahan_jari2_lobus }}</td>
        <td>{{ kanker.keparahan_tekstur_permukaan }}</td>
        <td>{{ kanker.keparahan_tumor_inti }}</td>
        <td>{{ kanker.keparahan_luas_lobus }}</td>
        <td>{{ kanker.keparahan_luas_permukaan_tumor }}</td>
        <td>{{ kanker.keparahan_cekungan_kontur }}</td>
        ...
        <td>{{ kanker.keparahan_jumlah_cekungan_kontur }}</td>
        <td>
          <a href="/edit/{{ kanker.id }}"><span class="btn btn-success">Edit</span></a>
          <a href="/delete/{{ kanker.id }}"><span class="btn btn-danger">Delete</span></a>
        </td>
      </tr>
    <% endfor %>
  </tbody>
</table>
```

C. URL

```
path('view/', views.view),
```

5.1.6.2 Halaman Tambah Data

A. Views

```
def knkr(request):
    if request.method == "POST":
        form = KankerForm(request.POST)
        if form.is_valid():
            try:
                form.save()
                return redirect('/view')
            except:
                pass
    else:
        form = KankerForm()
    return render(request, 'halambah.html', {'form': form})
```

B. Template

```
<div class="row justify-content-center">
    <div class="col-md-6">
        <div class="card shadow">
            <div class="card-header bg-dark">
                <div class="float-left">
                    <h4 class="m-0 text-light"><b>TAMBAH DATA</b></h4>
                </div>
                <div class="float-right">
                    <a href="#" class="btn btn-secondary btn-sm">Kembali</a>
                </div>
            </div>
            <div class="card-body bg-gray-100">
                <hr>
                <h4>Silahkan masukan data kanker.</h4>
                <hr>
                <form action="/knkr/" id="form-tambah" method="POST">
                    {% csrf_token %}
                    <div class="float-left">
                        <div class="form-row">
                            <div class="form-group col-md-15">
                                <label for=""><strong>Rerata Jari2 Lobus :</strong></label>
                                <div>{{ form.rerata_jari2_lobus }}</div>
                            </div>
                        </div>
                        <div class="form-row">
                            <div class="form-group col-md-15">
                                <label for=""><strong>Rerata Tumor Inti :</strong></label>
                                <div>{{ form.rerata_tumor_inti }}</div>
                            </div>
                        </div>
                    <div class="form-row">
```

```

<div class="form-group col-md-15">
    <label form=""><strong>Rerata Luas Lobus : </strong></label>
    <div>{{ form.rerata_luas_lobus }}</div>
</div>
<div class="form-row">
    <div class="form-group col-md-15">
        <label form=""><strong>SE Jari2 Lobus : </strong></label>
        <div>{{ form.se_jari2_lobus }}</div>
    </div>
</div>
<div class="form-row">
    <div class="form-group col-md-15">
        <label form=""><strong>SE Tekstur Permukaan : </strong></label>
        <div>{{ form.se_tekstur_permukaan }}</div>
    </div>
</div>
<div class="form-row">
    <div class="form-group col-md-15">
        <label form=""><strong>SE Tumor Inti : </strong></label>
        <div>{{ form.se_tumor_inti }}</div>
    </div>
</div>
<div class="form-row">
    <div class="form-group col-md-15">
        <label form=""><strong>SE Fraktal Spesimen : </strong></label>
        <div>{{ form.se_fraktal_spesimen }}</div>
    </div>
</div>
<div class="form-row">
    <div class="form-group col-md-15">
        <label form=""><strong>Keparahan Tekstur Permukaan : </strong></label>
        <div>{{ form.keparahan_tekstur_permukaan }}</div>
    </div>
</div>
<div class="form-row">
    <div class="form-group col-md-15">
        <label form=""><strong>Keparahan Tumor Inti : </strong></label>
        <div>{{ form.keparahan_tumor_inti }}</div>
    </div>
</div>
<div class="form-row">
    <div class="form-group col-md-15">
        <label form=""><strong>Keparahan Luas Lobus : </strong></label>
        <div>{{ form.keparahan_luas_lobus }}</div>
    </div>
</div>
</div>

<!-- bagian kiri -->
<div class="float-right">
    <div class="form-row">
        <div class="form-group col-md-15">
            <label form=""><strong>Rerata Luas Permukaan Tumor : </strong></label>
            <div>{{ form.rerata_luas_permukaan_tumor }}</div>
        </div>
    </div>
    <div class="form-row">
        <div class="form-group col-md-15">
            <label form=""><strong>Rerata Cekungan Kontur : </strong></label>

```

```

        <div>{{ form.rerata_cekungan_kontur }}</div>
    </div>
</div>
<div class="form-row">
    <div class="form-group col-md-15">
        <label for=" "><strong>Rerata Jumlah Cekungan Kontur : </strong></label>
        <div>{{ form.rerata_jumlah_cekungan_kontur }}</div>
    </div>
</div>
<div class="form-row">
    <div class="form-group col-md-15">
        <label for=" "><strong>SE Luas Permukaan Tumor : </strong></label>
        <div>{{ form.se_luas_permukaan_tumor }}</div>
    </div>
</div>
<div class="form-row">
    <div class="form-group col-md-15">
        <label for=" "><strong>SE Cekungan Kontur : </strong></label>
        <div>{{ form.se_cekungan_kontur }}</div>
    </div>
</div>
<div class="form-row">
    <div class="form-group col-md-15">
        <label for=" "><strong>SE Jumlah Cekungan Kontur : </strong></label>
        <div>{{ form.se_jumlah_cekungan_kontur }}</div>
    </div>
</div>
<div class="form-row">
    <div class="form-group col-md-15">
        <label for=" "><strong>Keparahan Jari2 Lobus : </strong></label>
        <div>{{ form.keparahan_jari2_lobus }}</div>
    </div>
</div>
<div class="form-row">
    <div class="form-group col-md-15">
        <label for=" "><strong>Keparahan Luas Permukaan Tumor : </strong></label>
        <div>{{ form.keparahan_luas_permukaan_tumor }}</div>
    </div>
</div>
<div class="form-row">
    <div class="form-group col-md-15">
        <label for=" "><strong>Keparahan Cekungan Kontur : </strong></label>
        <div>{{ form.keparahan_cekungan_kontur }}</div>
    </div>
</div>
<div class="form-row">
    <div class="form-group col-md-15">
        <label for=" "><strong>Keparahan Jumlah Cekungan Kontur : </strong></label>
        <div>{{ form.keparahan_jumlah_cekungan_kontur }}</div>
    </div>
</div>
<hr>
<div class="form-group">
    <button type="submit" class="btn btn-primary">Simpan</button>
    <button type="reset" class="btn btn-danger">Batal</button>
</div>
</div>
</form>
</div>
</div>

```

C. URL

```
path('knkr/', views.knkr),
```

5.1.6.3 Halaman Edit Data

A. Views

```
def edit(request, id):
    kanker = Kanker.objects.get(id=id)
    return render(request, 'edit.html', {'kanker': kanker})

def update(request, id):
    kanker = Kanker.objects.get(id=id)
    form = KankerForm(instance=kanker)

    if request.method == 'POST':
        form = KankerForm(request.POST, instance=kanker)
        if form.is_valid():
            form.save()
        return redirect('/view')
    return render(request, 'view.html', {'form': form})
```

B. Template

```
<div class="float-left">
    <h4 class="h4 m-0 text-light"><b>EDIT DATA</b></h4>
</div>
<div class="float-right">
    <a href="http://127.0.0.1:8000/view/" class="btn btn-secondary btn-sm">kembali</a>
</div>

<div class="card-body bg-gray-100">
    <h4>Silahkan edit data kanker.</h4>
    <hr>
    <form action="/update/{{kanker.id}}/" class="post-form" method="POST">
        <input type="hidden" name="id" id="id" required maxlength="20" value="{{ kanker.id }}"/>
        {{ csrf_token }}
        <div class="float-left">
            <div class="form-row">
                <div class="form-group col-md-15">
                    <label for="kode_produk"><strong>Rerata Jari2 Lobus :</strong></label>
                    <div>
                        <input type="number" class="form-control" name="rerata_jari2_lobus" value="{{ kanker.rerata_jari2_lobus }}"/>
                    </div>
                </div>
            <div class="form-row">
                <div class="form-group col-md-15">
                    <label for="jenis_produk"><strong>Rerata Tumor Inti :</strong></label>
                    <div>
                        <input type="number" class="form-control" name="rerata_tumor_inti" value="{{ kanker.rerata_tumor_inti }}"/>
                    </div>
                </div>
            </div>
        </div>
    </form>
</div>
```

```

</div>
<div class="form-row">
    <div class="form-group col-md-15">
        <label for="jenis_produk"><strong>Rerata Luas lobus : </strong></label>
        <div>
            <input type="number" class="form-control" name="rerata_luas_lobus" value="{{ kanker.rerata_luas_lobus }}"/>
        </div>
    </div>
<div class="form-row">
    <div class="form-group col-md-15">
        <label for="jenis_produk"><strong>SE Jari2 lobus : </strong></label>
        <div>
            <input type="number" class="form-control" name="se_jari2_lobus" value="{{ kanker.se_jari2_lobus }}"/>
        </div>
    </div>
<div class="form-row">
    <div class="form-group col-md-15">
        <label for="jenis_produk"><strong>SE Tekstur Permukaan : </strong></label>
        <div>
            <input type="number" class="form-control" name="se_tekstur_permukaan" value="{{ kanker.se_tekstur_permukaan }}"/>
        </div>
    </div>
<div class="form-row">
    <div class="form-group col-md-15">
        <label for="jenis_produk"><strong>SE Tumor Inti : </strong></label>
        <div>
            <input type="number" class="form-control" name="se_tumor_inti" value="{{ kanker.se_tumor_inti }}"/>
        <div>
            <input type="number" class="form-control" name="keparahan_tumor_inti" value="{{ kanker.keparahan_tumor_inti }}"/>
        </div>
    </div>
</div>


<div class="float-right">
    <div class="form-row">
        <div class="form-group col-md-15">
            <label for="jenis_produk"><strong>Rerata Luas Permukaan Tumor : </strong></label>
            <div>
                <input type="number" class="form-control" name="rerata_luas_permukaan_tumor" value="{{ kanker.rerata_luas_permukaan_tumor }}"/>
            </div>
        </div>
    </div>
    <div class="form-row">
        <div class="form-group col-md-15">
            <label for="jenis_produk"><strong>Rerata Cekungan Kontur : </strong></label>
            <div>
                <input type="number" class="form-control" name="rerata_cekungan_kontur" value="{{ kanker.rerata_ce_kontur }}"/>
            </div>
        </div>
    </div>
    <div class="form-row">
        <div class="form-group col-md-15">
            <label for="jenis_produk"><strong>Rerata Jumlah Cekungan Kontur : </strong></label>
            <div>
                <input type="number" class="form-control" name="rerata_jumlah_cekungan_kontur" value="{{ kanker.rerata_jumlah_ce_kontur }}"/>
            <div>
                ...
            </div>
        </div>
    </div>

```

```

        </div>
    </div>
<div class="form-row">
    <div class="form-group col-md-15">
        <label for="jenis_produk"><strong>SE Luas Permukaan Tumor : </strong></label>
        <div>
            <input type="number" class="form-control" name="se_luas_permukaan_tumor" value="{{ kanker.se_luas_
        </div>
    </div>
<div class="form-row">
    <div class="form-group col-md-15">
        <label for="jenis_produk"><strong>SE Cekungan Kontur : </strong></label>
        <div>
            <input type="number" class="form-control" name="se_cekungan_kontur" value="{{ kanker.se_cekungan_k
        </div>
    </div>
<div class="form-row">
    <div class="form-group col-md-15">
        <label for="jenis_produk"><strong>SE Jumlah Cekungan Kontur : </strong></label>
        <div>
            <input type="number" class="form-control" name="se_jumlah_cekungan_kontur" value="{{ kanker.se_jum
        </div>
    </div>
<div class="form-row">
    <div class="form-group col-md-15">
        <label for="jenis_produk"><strong>Keparahan Luas Lobus : </strong></label>
        <div>
            <input type="number" class="form-control" name="keparahan_luas_lobus" value="{{ kanker.keparahan_
        </div>
    </div>
<div class="form-row">
    <div class="form-group col-md-15">
        <label for="jenis_produk"><strong>Keparahan Luas Permukaan Tumor : </strong></label>
        <div>
            <input type="number" class="form-control" name="keparahan_luas_permukaan_tumor" value="{{ kanker.
        </div>
    </div>
<div class="form-row">
    <div class="form-group col-md-15">
        <label for="jenis_produk"><strong>Keparahan Cekungan Kontur : </strong></label>
        <div>
            <input type="number" class="form-control" name="keparahan_cekungan_kontur" value="{{ kanker.kepar
        </div>
    </div>
<div class="form-row">
    <div class="form-group col-md-15">
        <label for="jenis_produk"><strong>Keparahan Jumlah Cekungan Kontur : </strong></label>
        <div>
            <input type="number" class="form-control" name="keparahan_jumlah_cekungan_kontur" value="{{ kanke
        </div>
    </div>
<hr>
<div class="form-group">
    <button type="submit" class="btn btn-primary">Simpan</button>
    <button type="reset" class="btn btn-danger">Batal</button>
</div>
</div>
</form>

```

C. URL

```
path('edit/<int:id>', views.edit),
```

5.1.6.4 Hapus Data

A. Views

```
def delete(request, id):
    kanker = Kanker.objects.get(id=id)
    kanker.delete()
    return redirect("/view")
```

5.1.7 Implementasi Proses Menampilkan Visualisasi Data

A. Views

```
def visualisasi(request):
    return render(request, 'visualisasi.html')
```

B. Template

```
<div class='tableauPlaceholder' id='viz1673580729136' style='position: relative'>
  <noscript>
    <a href='#!'>
      |   <img alt='Dashboard 1' src='https://public.tableau.com/static/images/Diagnos...' />
    </a>
  </noscript>
  <object class='tableauviz' style='display:none;'>
    <param name='host_url' value='https%3A%2F%2Fpublic.tableau.com%2F' />
    <param name='embed_code_version' value='3' />
    <param name='site_root' value=''/>
    <param name='name' value='DiagnosisBreastCancer#47;Dashboard1' />
    <param name='tabs' value='no' />
    <param name='toolbar' value='yes' />
    <param name='static_image' value='https://public.tableau.com/static/images/Diagnos...' />
    <param name='animate_transition' value='yes' />
    <param name='display_static_image' value='yes' />
    <param name='display_spinner' value='yes' />
    <param name='display_overlay' value='yes' />
    <param name='display_count' value='yes' />
    <param name='language' value='en-US' />
    <param name='filter' value='publishedyes' />
  </object>
</div>
<script type='text/javascript'>
var divElement = document.getElementById('viz1673580729136');
var vizElement = divElement.getElementsByTagName('object')[0];
if ( divElement.offsetWidth > 800 )
{
  { vizElement.style.minWidth='220px';vizElement.style.maxWidth='590px';vizElement.style.width='100%' ;vizElement.style.height='727px' }
  else if ( divElement.offsetWidth > 500 )
  { vizElement.style.minWidth='220px';vizElement.style.maxWidth='590px';vizElement.style.width='100%' ;vizElement.style.height='727px' }
}
else
{ vizElement.style.width='100%';vizElement.style.height='727px';}
var scriptElement = document.createElement('script');
scriptElement.src = 'https://public.tableau.com/javascripts/api/viz_v1.js';
vizElement.parentNode.insertBefore(scriptElement, vizElement);
</script>
```

C. URL

```
path('visualisasi', views.visualisasi, name='visualisasi'),
```

5.1.8 Implementasi Proses Logout

A. Views

```
51     def LogoutPage(request):
52         logout(request)
53         return redirect('login')
54
```

5.2 Implementasi Halaman Antar Muka

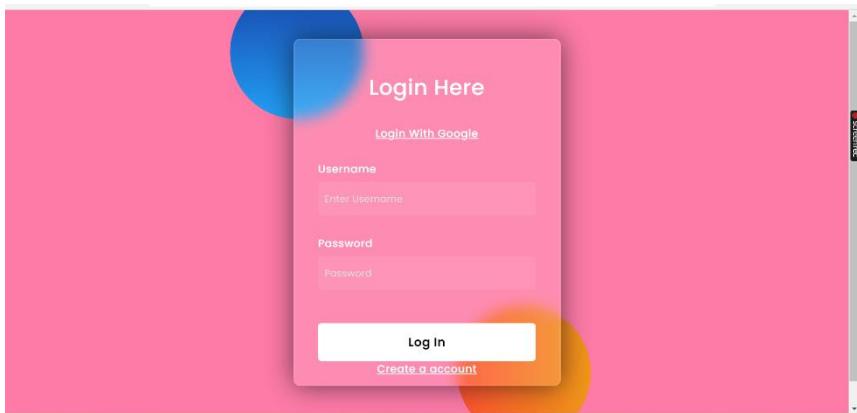
5.2.1. Implementasi Halaman Depan



Gambar 5. 1 Halaman Depan Aplikasi

Gambar diatas merupakan implementasi dari halaman depan sistem. Pada halaman ini, *user* dapat menekan button “*Login*” pada navbar untuk masuk ke dalam sistem.

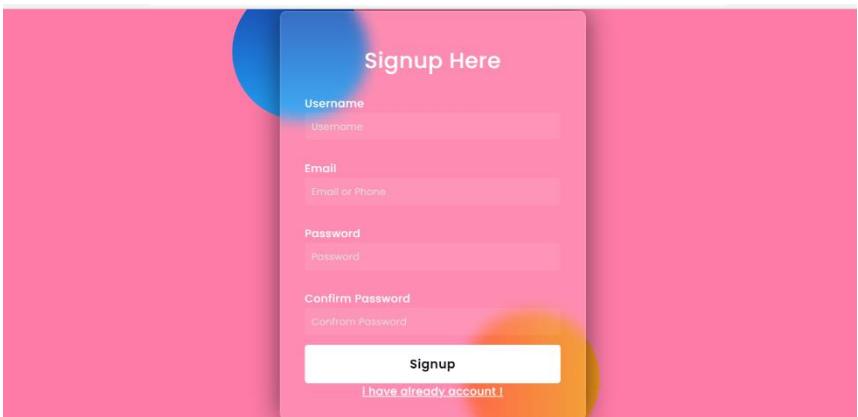
5.2.2. Implementasi Halaman Login



Gambar 5. 2 Halaman Login Aplikasi

Gambar diatas merupakan implementasi dari halaman *login*. Sebelum masuk ke halaman *dashboard*, *user* harus menginputkan *username* dan *password* yang sesuai. Jika sesuai, maka *user* akan di arahkan ke halaman *dashboard*. Jika tidak *user* akan diminta kembali memasukkan *username* dan *password*yang sesuai.

5.2.3. Implementasi Halaman Registrasi



Gambar 5. 3 Halaman Registrasi Aplikasi

Gambar diatas merupakan implementasi dari halaman registrasi. Jika *user* belum memiliki akun untuk masuk sistem, maka *user* dapat melakukan registrasi dengan cara menginputkan *username* dan *password* lalu menekan tombol register.

5.2.4. Implementasi Halaman Prediksi Kanker Payudara

The screenshot shows a web-based application for predicting breast cancer. At the top, there is a navigation bar with links to Home, Predict, Visualization, and Logout. Below the navigation bar is a title "Aplikasi Prediksi Kanker Payudara" and a subtitle "Silahkan Inputkan data Anda !". The main area contains a grid of input fields for various parameters. The parameters are grouped into two columns:

Rerata Jari-Jari Lobus :	Rerata Tumor Inti :
Masukan Rerata Jari-Jari Lobus	Masukan Rerata Tumor Inti
Rerata Luas Lobus :	Rerata Luas Permukaan Tumor :
Masukan Rerata Luas Lobus	Masukan Rerata Luas Permukaan Tumor
Rerata Cekungan Kontur :	Rerata Jumlah Cekungan Kontur :
Masukan Rerata Cekungan Kontur	Masukan Rerata Jumlah Cekungan Kontur
SE Jari-jari Lobus :	Tekstur Permukaan :
Masukan SE jari-jari Lobus	Masukan SE Tekstur Permukaan
SE Tumor Inti :	SE Luas Permukaan Tumor :
Masukan SE Tumor Inti	Masukan SE Luas Permukaan Tumor
SE Cekungan Kontur :	SE Jumlah Cekungan Kontur :
Masukan SE Cekungan Kontur	Masukan SE Jumlah Cekungan Kontur
SE Fraktal Specimen :	Keparahan Jari-Jari Lobus :
Masukan SE Fraktal Specimen	Masukan Keparahan Jari-Jari Lobus
Keparahan Tekstur Permukaan :	Keparahan Tumor Inti :
Masukan Keparahan Tekstur Permukaan	Masukan Keparahan Tumor Inti
Keparahan Luas Lobus :	Keparahan Luas Permukaan Tumor :
Masukan Keparahan Luas Lobus	Masukan Keparahan Luas Permukaan Tumor
Keparahan Cekungan Kontur :	Keparahan Jumlah Cekungan Kontur :
Masukan Keparahan Cekungan Kontur	Masukan Keparahan Jumlah Cekungan Kontur

At the bottom center of the form is a button labeled "Lakukan Prediksi".

Gambar 5. 4 Halaman Prediksi Kanker

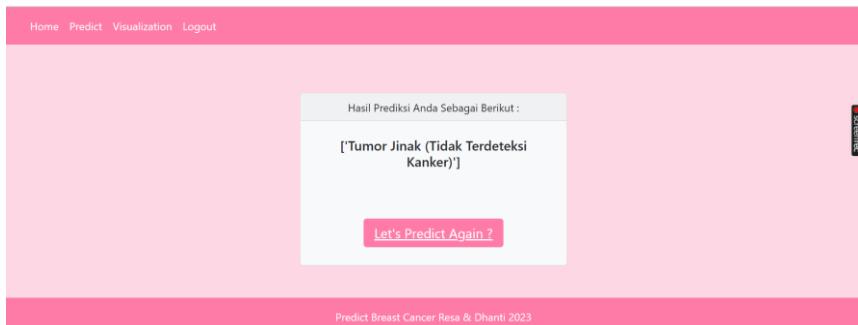
Gambar diatas merupakan implementasi dari halaman prediksi dari sistem yang dirancang. Pada halaman ini, terdapat *form* untuk melakukan prediksi kanker payudara. *User* dapat menginputkan angka ke dalam *form* untuk mengetahui hasil prediksi kanker payudara. *Field* tersebut diantaranya adalah sebagai berikut.

1. Rerata Jari-jari Lobus
2. Rerata Tumor Inti
3. Rerata Luas Lobus

4. Rerata Luas Permukaan Tumor
5. Rerata Cekungan Kontur
6. Rerata Jumlah Cekungan Kontur
7. SE Jari-jari Lobus
8. SE Tekstur Permukaan
9. SE Tumor Inti
10. SE Luas Permukaan Tumor
11. SE Cekungan Kontur
12. SE Jumlah Cekungan Kontur
13. Keparahan Jari-jari Lobus
14. Keparahan Tekstur Permukaan
15. Keparahan Tumor Inti
16. Keparahan Luas Lobus
17. Keparahan Luas Permukaan Tumor
18. Keparahan Cekungan Kontur
19. Keparahan Jumlah Cekungan Kontur

Kemudian *user* dapat menekan *button* “Lakukan Prediksi” untuk melihat hasil prediksi berdasarkan data yang diinputkan.

5.2.5. Implementasi Halaman Hasil Prediksi Kanker Payudara



Gambar 5. 5 Halaman Hasil Prediksi Aplikasi

Gambar diatas merupakan implementasi dari halaman hasil prediksi dari aplikasi yang dirancang. Pada halaman ini, akan ditampilkan hasil prediksi kanker payudara dari *user* yang sebelumnya telah menginputkan nilai pada *form* prediksi.

5.2.6. Implementasi Halaman Menampilkan Data Kanker Payudara

List Cancer Datas																		[+] Tambah Data Kanker																	
Id	Rerata Jari2	Rerata Tumor	Rerata Luas	Rerata Rata2	Rerata Luas	Rerata Cekungan	Rerata Kontur	Rerata Jumlah	SE Jari2	SE Tekstur	SE Tumor	SE Luas	SE Inti	SE Cekungan	SE Jumlah	SE Fraktal	Keparahan Jari2	Keparahan Tekstur	Keparahan Spesimen	Keparahan Tumor	Keparahan Inti	Keparahan Luas	Keparahan Lobus	Kepu L	Kepu Perm Tu										
	1	101	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100										

Gambar 5. 6 Halaman Data Kanker Payudara

Gambar diatas merupakan implementasi dari halaman data kanker payudara dari aplikasi yang dirancang. Pada halaman ini, akan ditampilkan data dalam bentuk tabel yang terdiri dari atribut sebagai berikut.

1. Rerata Jari-jari Lobus
2. Rerata Tumor Inti

3. Rerata Luas Lobus
4. Rerata Luas Permukaan Tumor
5. Rerata Cekungan Kontur
6. Rerata Jumlah Cekungan Kontur
7. SE Jari-jari Lobus
8. SE Tekstur Permukaan
9. SE Tumor Inti
10. SE Luas Permukaan Tumor
11. SE Cekungan Kontur
12. SE Jumlah Cekungan Kontur
13. Keparahan Jari-jari Lobus
14. Keparahan Tekstur Permukaan
15. Keparahan Tumor Inti
16. Keparahan Luas Lobus
17. Keparahan Luas Permukaan Tumor
18. Keparahan Cekungan Kontur
19. Keparahan Jumlah Cekungan Kontur

Record yang ditampilkan terdiri dari 569 *record*. Terdapat button Tambah Data yang berfungsi untuk menambah data pegawai, button Edit berfungsi untuk mengubah suatu *record* yang dipilih, dan button *Delete* untuk menghapus suatu *record* yang dipilih.

5.2.7. Implementasi Halaman Tambah Data Kanker Payudara

TAMBAH DATA

Silahkan masukan data kanker.

Rerata Jari2 Lobus :

Rerata Luas Permukaan Tumor :

Rerata Tumor Inti :

Rerata Cekungan Kontur :

Rerata Luas Lobus :

Rerata Jumlah Cekungan Kontur :

SE Jari2 Lobus :

SE Luas Permukaan Tumor:

SE Tekstur Permukaan :

SE Cekungan Kontur :

SE Tumor Inti :

SE Jumlah Cekungan Kontur :

SE Fraktal Spesimen :

Keparahan Jari2 Lobus :

Keparahan Tekstur Permukaan :

Keparahan Luas Permukaan Tumor :

Keparahan Tumor Inti :

Keparahan Cekungan Kontur :

Keparahan Luas Lobus :

Keparahan Jumlah Cekungan Kontur :

Gambar 5. 7 Halaman Tambah Data Kanker Payudara Aplikasi

Gambar diatas merupakan implementasi dari halaman tambah data kanker payudara dari aplikasi yang dirancang. Pada halaman ini, akan ditampilkan *form* tambah data untuk menambah data pegawai. *User* dapat menginputkan data-data yang ada pada atribut data kanker payudara pada *form*. Lalu *user* dapat menekan button Simpan untuk menyimpan data yang diinputkan. *User* dapat menekan button Batal untuk *reset* data yang diinputkan pada *form*.

5.2.8. Implementasi Halaman Edit Data Kanker Payudara

EDIT DATA

Silahkan edit data kanker.

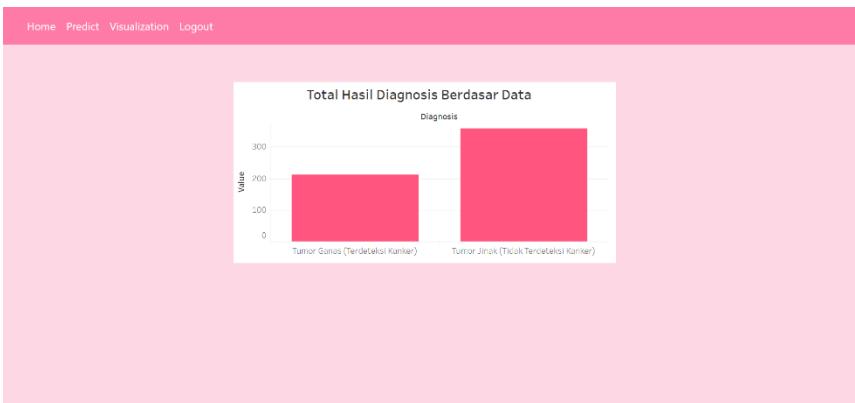
Rerata Jari2 Lobus :	100	Rerata Luas Permukaan Tumor :	100
Rerata Tumor Inti :	100	Rerata Cekungan Kontur :	100
Rerata Luas Lobus :	100	Rerata Jumlah Cekungan Kontur :	100
SE Jari2 Lobus :	100	SE Luas Permukaan Tumor :	100
SE Tekstur Permukaan :	100	SE Cekungan Kontur :	100
SE Tumor Inti :	100	SE Jumlah Cekungan Kontur :	100
SE Fraktal Spesimen :	100	Keparahan Luas Lobus :	100
Keparahan Jari2 Lobus :	100	Keparahan Luas Permukaan Tumor :	100
Keparahan Tekstur Permukaan :	100	Keparahan Cekungan Kontur :	100
Keparahan Tumor Inti :	100	Keparahan Jumlah Cekungan Kontur :	100

Simpan Batal

Gambar 5. 8 Halaman Edit Data Kanker Payudara

Gambar diatas merupakan implementasi dari halaman edit data kanker payudara dari aplikasi yang dirancang. Pada halaman ini, akan ditampilkan *form* edit data untuk mengubah data kanker payudara. *User* dapat mengubah nilai dari atribut yang dipilih pada data *form*. Lalu *user* dapat menekan button Simpan untuk menyimpan data yang telah diubah. *User* dapat menekan button Batal untuk me-reset data yang diinputkan pada *form*.

5.2.9. Implementasi Halaman Visualisasi



Gambar 5. 9 Halaman Visualisasi Aplikasi

Gambar diatas merupakan implementasi dari halaman visualisasi dari aplikasi yang dirancang. Pada halaman ini, akan ditampilkan bentuk visualisasi diagram batang total jumlah kanker dan tidak kanker pada data yang diinputkan.

BAB 6

Kesimpulan dan Saran

6.1 Kesimpulan

Berdasarkan hasil analisis dan pembahasan yang telah dilakukan dapat disimpulkan beberapa poin yaitu :

1. Model prediksi yang dirancang menggunakan algoritma Support Vector Machine yang dibuat berhasil melakukan prediksi kanker payudara dengan baik, sehingga dapat disimpulkan bahwa model yang dibangun memiliki performa yang baik untuk memprediksi kanker payudara dengan menggunakan sembilan belas variabel independen.
2. Berdasarkan model yang dibuat didapatkan nilai akurasi sebesar 98.24%. Akurasi tersebut merupakan nilai akurasi yang baik, sehingga dapat dikatakan model *machine learning* dapat berperforma baik untuk memprediksi kanker payudara.
3. Berdasarkan uji validitas, nilai akurasi 0,9824 menunjukkan bahwa Diagnosis dipengaruhi oleh faktor independen 98.24%. Nilai sisa dari akurasi tersebut adalah 0,176 yang artinya Diagnosis dipengaruhi oleh faktor lain yang tidak diketahui sebesar 1.76%.
4. Visualisasi data dari hasil model prediksi kanker payudara dapat digunakan menjadi bentuk aplikasi berbasis web dengan menggunakan *framework* Django. Dengan aplikasi tersebut, admin dapat melakukan prediksi kanker payudara dengan mudah dan dengan cepat.

6.2 Saran

Saran yang dapat disampaikan pada peneliti yang akan melanjutkan dan mengembangkan penelitian ini adalah :

1. Sumber data yang digunakan perlu dikombinasikan dengan data yang berasal dari Rumah Sakit untuk melakukan testing model prediksi.
2. Pada halaman visualisasi, diagram batang belum mampu berubah sesuai data yang diinputkan, untuk itu pengembangan dapat dilakukan dengan membuat grafik sesuai dengan data yang diinputkan.

DAFTAR PUSTAKA

- [1] Lina Mardiana, *Kanker pada Wanita*. Penebar Swadaya, 2007. Accessed: Jan. 11, 2023. [Online]. Available: <https://books.google.co.id/books?id=IYCN1e9JtQcC&printsec=frontcover&hl=id#v=onepage&q&f=false>
- [2] Sitiatava Rizema Putra, *Buku Lengkap Kanker Payudara*. Laksana, 2015. Accessed: Jan. 11, 2023. [Online]. Available: https://books.google.co.id/books?hl=id&lr=&id=LdpwEA AAQBAJ&oi=fnd&pg=PA5&dq=payudara+adalah&ots=bxg-cPVm1n&sig=QL-CzCt7Jcmsv31wg1_fwszA_Nc&redir_esc=y#v=onepage&q=payudara%20adalah&f=false
- [3] A. I. S. Azis, I. Surya Kumala Idris, B. Santoso, and Y. Aril Mustofa, "Pendekatan Machine Learning yang Efisien untuk Prediksi Kanker Payudara," *Jurnal Rekayasa Sistem dan Teknologi Informasi*, vol. 3, no. 3, pp. 458-469, 2019, Accessed: Oct. 22, 2022. [Online]. Available: <http://jurnal.iaii.or.id/index.php/RESTI/article/view/1347/180>
- [4] A. Bharat, N. Pooja, and R. A. Reddy, "Using Machine Learning algorithms for breast cancer risk prediction and diagnosis," *IEEE Third International Conference on Circuits, Control, Communication and Computing*, 2018, doi: <https://doi.org/10.1109/CIMCA.2018.8739696>.
- [5] dr. Rizal Fadli, "Kanker Payudara," 2022. <https://www.halodoc.com/kesehatan/kanker-payudara> (accessed Jan. 11, 2023).
- [6] World Health Organization, "Breast Cancer," 2021. <https://www.who.int/news-room/fact-sheets/detail/breast-cancer> (accessed Jan. 11, 2023).
- [7] Doktersehat.com, "12 Cara Mencegah Kanker Payudara

- yang Perlu Wanita Ketahui!,” 2020. <https://rsud.sawahluntokota.go.id/12-cara-mencegah-kanker-payudara-yang-perlu-wanita-ketahui/> (accessed Jan. 11, 2023).
- [8] World Cancer Research Data Cancer, “Worldwide cancer data,” 2022. <https://www.wcrf.org/cancer-trends/worldwide-cancer-data/> (accessed Jan. 11, 2023).
- [9] Nizar Rabbi Radliya, *Data Mining*. 2015.
- [10] Samsudiney, “Penjelasan Sederhana tentang Apa Itu SVM?,” 2019. <https://medium.com/@samsudiney/penjelasan-sederhana-tentang-apa-itu-svm-149fec72bd02> (accessed Jan. 11, 2023).
- [11] O. Pahlevi, A. Mulyani, and M. Khoir, “Sistem Informasi Inventory Barang Menggunakan Metode Object Oriented Di PT. Livaza Teknologi Indonesia Jakarta,” *Jurnal PROSISKO*, vol. 5, no. 1, pp. 27-35, 2018, Accessed: Jun. 16, 2022. [Online]. Available: <https://ejurnal.ippmunsera.org/index.php/PROSISKO/article/view/587>
- [12] D. Saputra and R. Fathoni Aji, “Analisis Perbandingan Performa Web Service Rest Menggunakan Framework Laravel, Django Dan Ruby On Rails Untuk Akses Data Dengan Aplikasi Mobile (Studi Kasus: Portal E-Kampus STT Indonesia Tanjungpinang),” *Bangkit Indonesia*, vol. 2, no. 2, pp. 17-22, 2018, doi: 10.52771/bangkitindonesia.v7i2.90.
- [13] A. N. Rahimah, D. S. Rusdianto, and M. T. Ananta, “Pengembangan Sistem Pengelolaan Ruang Baca Berbasis Web Dengan Menggunakan Django Framework (Studi Kasus: Ruang Baca Fakultas Ilmu Komputer Universitas Brawijaya),” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 3, no. 5, pp. 4439-4446, 2019, Accessed: Jun. 16, 2022. [Online]. Available: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/5227>
- [14] Hasan Bisri Isa Alfaris, Choirul Anam, and Ali Masy'an,

- “Implementasi Black Box Testing Pada Sistem Informasi Pendaftaran Santri Berbasis Web Dengan Menggunakan PHP Dan MySQL,” *Jurnal Sains dan Teknologi*, vol. 6, no. 1, pp. 23-38, 2013, doi: 10.32764/saintekbu.v6i1.64.
- [15] Yudo Ardianto Agung, “Apa itu VsCode?,” 2020. <https://rep.alphabetincubator.id/apa-itu-vscode/> (accessed Jun. 16, 2022).
 - [16] small-business-tracker.com, “Apa itu Jupyter Notebook? Analisis data menjadi lebih mudah,” 2022. <https://ind.small-business-tracker.com/what-is-jupyter-notebook-data-analysis-made-easier-574170> (accessed Jun. 16, 2022).
 - [17] Institut Teknologi Bandung, “Tableau,” 2022. <https://ditsti.itb.ac.id/en/tableau/> (accessed Jun. 13, 2022).
 - [18] IndoML, “Panduan Menggunakan Kaggle untuk Pemula,” *IndoML.com*, 2017. <https://indoml.com/2017/08/22/panduan-menggunakan-kaggle-untuk-pemula/> (accessed Jan. 11, 2023).
 - [19] E. W. Fridayanthie and T. Mahdiati, “Rancang Bangun Sistem Informasi Permintaan Atk Berbasis Intranet (Studi Kasus: Kejaksaan Negeri Rangkasbitung),” *Jurnal Khatulistiwa Informatika*, vol. 4, no. 2, pp. 126-138, 2016, doi: 10.31294/jki.v4i2.1264.

-oo00oo-

BIODATA PENULIS



Cahyo Prianto, S.Pd., M.T., CDSP, SFPC. Lulus S1 di Program Studi Pendidikan Fisika Universitas Pendidikan Indonesia tahun 2007, dan lulus S2 di Program Studi Teknik Elektro Institut Teknologi Bandung tahun 2010. Saat ini adalah dosen tetap Program Studi Teknik Informatika di Universitas Logistik dan Bisnis Internasional. Mengampu matakuliah Matematika Diskrit, Aljabar Linier, dan Data Mining. Aktif menulis di berbagai jurnal ilmiah.



Resa Rianti. Lahir di Bandung, 23 September. Saat ini adalah mahasiswa Program Studi D4 Teknik Informatika di Universitas Logistik dan Bisnis Internasional.



Nur Tri Ramadhanti Adiningrum. Lahir di Bandung, 06 Desember 2001. Saat ini adalah mahasiswa Program Studi D4 Teknik Informatika di Universitas Logistik dan Bisnis Internasional.

MEMBUAT APLIKASI PREDIKSI KANKER PAYUDARA TANPA PUSING

KANKER PAYUDARA MERUPAKAN PENYAKIT YANG PALING BANYAK DITEMUKAN PADA WANITA DAN ANGKA KEMATIANNYA MASIH MENEMPATI POSISI KEDUA KASUS KANKER YANG DAPAT MEMPENGARUHI LEBIH DARI 2,1 JUTA ORANG PADA TAHUN 2015. BERDASAR KASUS TERSEBUT DAPAT DIKETAHUI BAHWA KANKER PAYUDARA ADALAH JENIS KANKER YANG MENJADI PENYEBAB UTAMA KEMATIAN WANITA, NAMUN KEMATIAN INI DAPAT DIKURANGI DENGAN MELAKUKAN DETEKSI DINI TERHADAP SEL KANKER. OLEH KARENA ITU, DETEKSI KANKER BERPERAN PENTING DALAM PROSES PENGOBATAN DAN MEMBANTU MENINGKATKAN ANGKA KELANGSUNGAN HIDUP. PREDIKSI KANKER DAPAT MEMBANTU PASIEN UNTUK BERKONSULTASI DENGAN DOKTER LEBIH CEPAT. SEHINGGA, DENGAN PREDIKSI KANKER YANG TEPAT SANGAT PENTING UNTUK MEMPERBARUI PERAWATAN PASIEN KANKER PAYUDARA. TEKNIK MACHINE LEARNING DAPAT DILAKUKAN UNTUK MEMPREDIKSI KANKER PAYUDARA KARENA DAPAT MENANGKAP INTERAKSI TINGKAT TINGGI ANTAR DATA YANG MUNGKIN MENGHASILKAN PREDIKSI YANG LEBIH BAIK UNTUK MEMBEDAKAN ANTARAJINAK DAN GANAS. OLEH KARENA ITU, PENELITIAN INI MEMPERKENALKAN PENDEKATAN KLASIFIKASI YANG EFektif BERDASARKAN SUPPORT VECTOR MACHINE. SUPPORT VECTOR MACHINE (SVM) ADALAH MODEL YANG DIGUNAKAN UNTUK MELAKUKAN PREDIKSI KANKER PAYUDARA. UNTUK MEMpermudah PROSES PREDIKSI, HASIL PREDIKSI KANKER PAYUDARA DITERAPKAN DALAM BENTUK WEB BASE DENGAN FRAMEWORK DJANGO UNTUK DAPAT DIGUNAKAN OLEH DOKTER DALAM MENENTUKAN KEPUTUSAN DENGAN CEPAT. HASIL MODELING MENUNJUKKAN BAHWA PREDIKSI KANKER PAYUDARA MENGGUNAKAN SVM MEMPEROLEH AKURASI TERTINGGI YAITU 98,24%.



Cahyo Prianto, S.Pd., M.T.,CDSP, SFPC. Lulus S1 di Program Studi Pendidikan Fisika Universitas Pendidikan Indonesia tahun 2007, dan lulus S2 di Program Studi Teknik Elektro Institut Teknologi Bandung tahun 2010. Saat ini adalah dosen tetap Program Studi Teknik Informatika di Universitas Logistik dan Bisnis Internasional. Mengampu matkulah Matematika Diskrit, Aljabar Linier, dan Data Mining. Aktif menulis di berbagai jurnal ilmiah.



Resa Rianti. Saat ini adalah Mahasiswa Program Studi D4 Teknik Informatika di Universitas Logistik dan Bisnis Internasional.



Nur Tri Ramadhanti Adiningrum. Lahir di Bandung, 06 Desember 2001. Saat ini adalah mahasiswa Program Studi D4 Teknik Informatika di Universitas Logistik dan Bisnis Internasional.