

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian tugas akhir ini memerlukan studi literatur dalam bentuk beberapa karya ilmiah seperti, jurnal dan skripsi sebagai bahan pertimbangan untuk memperkuat bahwa penelitian ini merupakan penelitian asli serta dapat membedakan penelitian ini dengan penelitian terdahulu. Jurnal dan skripsi yang dijadikan referensi berkaitan dengan tema penelitian yaitu perancangan pengembangan *website* E-Posyandu dengan pendekatan *machine learning* algoritma KNN.

Penelitian pertama tentang perancangan dan implementasi e-posyandu berbasis web yang dilakukan pada tahun 2022. Studi ini memiliki tujuan untuk mengembangkan dan memplementasikan situs web e-posyandu dengan tujuan utama untuk memperbaiki kualitas layanan yang diberikan oleh para kader posyandu dalam proses komunikasi hasil penilaian pertumbuhan dan perkembangan anak balita kepada para orang tua. Pada pengembangan *website* ini dibangun dengan bahasa pemrograman PHP *native*. Tetapi pada *website* ini tidak ada sistem deteksi *stunting* secara otomatis[6].

Penelitian kedua tentang perbandingan performa metode KNN dan *Naïve Bayes* pada klasifikasi status gizi anak balita yang dilakukan pada tahun 2022. Objektif dari studi ini adalah untuk melakukan evaluasi komparatif terhadap efisiensi algoritma *K-Nearest Neighbors* (KNN) dibandingkan dengan *Naïve Bayes* dalam mengklasifikasikan status nutrisi pada anak balita. Hasil dari investigasi menunjukkan bahwa, ketika membandingkan efektivitas kedua metode tersebut dengan menggunakan skor F1 sebagai kriteria utama penilaian performa klasifikasi, algoritma *K-Nearest Neighbors* menunjukkan keunggulan yaitu *f1-score* yang dicapai adalah 45.95%, dengan perbedaan mencapai 13,42% pada *f1-score*[5]. Implikasi dari temuan ini menegaskan dominasi

metode *K-Nearest Neighbors* atas *Naïve Bayes Classifier* dalam konteks klasifikasi status nutrisi pada anak balita.

Penelitian ketiga tentang klasifikasi kNN yang efisien dengan jumlah tetangga terdekat yang berbeda yang dilakukan pada tahun 2018. Dalam penelitian ini, peneliti telah mengusulkan dua algoritma klasifikasi kNN baru, yaitu metode kTree dan k*Tree, untuk memilih nilai-k yang optimal untuk setiap sampel uji untuk klasifikasi kNN yang efisien dan efektif. Ide utama dari metode yang peneliti usulkan adalah untuk merancang tahap pelatihan untuk mengurangi biaya operasional tahap pengujian dan meningkatkan kinerja klasifikasi. Dua set percobaan telah dilakukan untuk mengevaluasi metode yang diusulkan dengan metode yang bersaing, dan hasil percobaan menunjukkan bahwa metode peneliti mengungguli metode yang bersaing dalam hal akurasi klasifikasi dan biaya operasional[7].

Penelitian keempat tentang penerapan algoritma *Support Vector Machine* (SVM) dan *K-Nearest Neighbor* (KNN) dengan teknik seleksi fitur *Backward Elimination* (BE) untuk memprediksi kejadian *stunting* di kalangan anak balita, yang dilaksanakan pada tahun 2022. Tujuan utama dari penelitian adalah untuk mengkaji keberhasilan dan keakuratan algoritma SVM dan KNN yang diintegrasikan dengan metode BE dalam menghasilkan model prediktif yang optimal. Berdasarkan eksperimen yang telah dilakukan, model prediktif untuk mengestimasi prevalensi *stunting* dengan menggunakan kedua algoritma tersebut telah berhasil diimplementasikan. Temuan ini menawarkan wawasan yang bisa dipertimbangkan untuk pengembangan kebijakan atau dalam proses pengambilan keputusan terkait. Secara khusus, dalam konteks kesalahan prediksi, algoritma KNN menunjukkan nilai kesalahan *Root Mean Square Error* (RMSE) yang paling rendah, yaitu 2,476, sebuah nilai yang lebih baik bila dibandingkan dengan hasil dari algoritma SVM[8].

Penelitian kelima tentang penilaian status *stunting* pada anak-anak melalui penerapan algoritma *K-Nearest Neighbors* (KNN) yang dijalankan pada tahun 2021. Tujuan dari penelitian ini adalah untuk mengklasifikasikan status

nutrisi pada anak usia dini menggunakan algoritma KNN serta untuk mengevaluasi tingkat kepresisian yang dicapai oleh algoritma tersebut dalam klasifikasi tersebut. Hasil dari penelitian ini mengindikasikan bahwa keakuratan paling tinggi dan laju kesalahan paling rendah tercapai pada parameter k sebesar 3, dengan keakuratan mencapai 83% dan laju kesalahan sebesar 0,142[9].

Tabel 2.1 Penelitian terkait klasifikasi *stunting machine learning* algoritma KNN dan pengembangan *website* E-Posyandu.

No	Peneliti	Judul	Pendekatan Metode	Hasil
1	Muhasshanah M, Abd. Ghofur, Fatimatuzzahra F. (2022)	Perancangan dan implementasi e-posyandu untuk peningkatan pelayanan kader di posyandu delima berbasis web.	Penerapan metode <i>waterfall</i>	<i>Website</i> ini dapat digunakan oleh kader sebagai pengelola untuk menginputkan data ibu, balita, hasil pemeriksaan balita. Tetapi pada <i>website</i> ini tidak ada sistem deteksi <i>stunting</i> secara otomatis[6].
2	Moch. Rizky, B Muslimin, Franz A (2022)	Perbandingan Metode <i>K-Nearest Neighbors</i> dan <i>Naïve Bayes Classifier</i> Pada Klasifikasi Status Gizi Balita di Puskesmas Muara Jawa Kota Samarinda	Perbandingan <i>K-Nearest Neighbor</i> dan <i>Naïve Bayes</i>	Objektif dari studi ini adalah untuk melakukan evaluasi komparatif terhadap efisiensi algoritma <i>K-Nearest Neighbors</i> (KNN) dibandingkan dengan <i>Naïve Bayes</i> dalam mengklasifikasikan status nutrisi pada anak balita. Hasil dari investigasi menunjukkan bahwa, ketika membandingkan efektivitas kedua metode tersebut dengan menggunakan skor F1 sebagai kriteria utama penilaian performa klasifikasi, algoritma <i>K-Nearest Neighbors</i> menunjukkan keunggulan yang signifikan, dengan perbedaan mencapai 13,42% pada skor F1[5].
3	Zhang S, Li X, Zong M et al. (2018)	<i>Efficient kNN Classification With Different Numbers of Nearest Neighbors</i>	<i>K-Nearest Neighbor (KNN)</i>	Dalam penelitian ini, peneliti telah mengusulkan dua algoritma klasifikasi kNN baru, yaitu metode kTree dan k*Tree, untuk memilih nilai-k yang optimal untuk setiap sampel uji untuk klasifikasi kNN

				yang efisien dan efektif. Ide utama dari metode yang peneliti usulkan adalah untuk merancang tahap pelatihan untuk mengurangi biaya operasional tahap pengujian dan meningkatkan kinerja klasifikasi. Dua set percobaan telah dilakukan untuk mengevaluasi metode yang diusulkan dengan metode yang bersaing, dan hasil percobaan menunjukkan bahwa metode peneliti mengungguli metode yang bersaing dalam hal akurasi klasifikasi dan biaya operasional[7].
4	Labolo A, Mooduto S, Bode A. (2022)	Penerapan Algoritma <i>Support Vector Machine</i> dan <i>K-Nearest Neighbor</i> Menggunakan <i>Feature Selection Backward Elimination</i> Untuk Prediksi Status Penderita <i>Stunting</i> Pada Balita	<i>Support Vector Machine (SVM)</i> dan <i>K-Nearest Neighbor (KNN)</i>	Menurut hasil pengujian tingkat gagal terkecil dalam konteks kesalahan prediksi, algoritma KNN menunjukkan nilai kesalahan <i>Root Mean Square Error (RMSE)</i> yang paling rendah, yaitu 2,476, sebuah nilai yang lebih baik bila dibandingkan dengan hasil dari algoritma SVM. Hasil yang didapat dari prediksi menghasilkan nilai keakuratan 88,46%. Tetapi pada penelitian ini tidak adanya kelanjutan untuk mengintegrasikan dengan aplikasi atau sistem lain[8].
5	Saeful Bachri O, Mohamad R, Bhakti H. (2021)	Penentuan Status <i>Stunting</i> Pada Anak Dengan Menggunakan Algoritma KNN	<i>K-Nearest Neighbor (KNN)</i>	Menurut hasil dari pengujian yang telah dilakukan bahwa tingkat akurasi tertinggi dan nilai <i>error rate</i> terkecil yang didapatkan terdapat pada nilai k=3 yaitu 83% dengan <i>error rate</i> 0,142. Tetapi pada penelitian ini tidak adanya kelanjutan untuk mengintegrasikan dengan aplikasi atau sistem lain[9].

Penelitian pertama dilakukan pada tahun 2022 dan bertujuan untuk mengembangkan situs *web* e-posyandu berbasis *web*, fokus pada meningkatkan kualitas layanan kader posyandu dalam komunikasi hasil penilaian pertumbuhan anak balita kepada orang tua. Penelitian kedua, juga tahun 2022, membandingkan efisiensi algoritma *K-Nearest Neighbors* (KNN) dengan *Naïve Bayes* dalam mengklasifikasikan status gizi anak balita, dengan hasil menunjukkan dominasi KNN dengan *f1-score* 45,95%. Penelitian ketiga pada 2018 mengusulkan algoritma klasifikasi KNN baru (kTree dan k*Tree) untuk klasifikasi yang efisien dan efektif dengan hasil mengungguli metode bersaing dalam akurasi klasifikasi dan biaya operasional. Penelitian keempat pada 2022 mempertimbangkan penerapan algoritma *Support Vector Machine* (SVM) dan KNN dengan teknik seleksi fitur *Backward Elimination* (BE) untuk memprediksi kejadian *stunting*, menunjukkan bahwa KNN memiliki nilai RMSE lebih rendah. Penelitian kelima pada tahun 2021 fokus pada penilaian status *stunting* anak-anak dengan menggunakan algoritma KNN, mencapai tingkat keakuratan tertinggi (83%) pada parameter $k=3$ dengan laju kesalahan 0,142.

2.2 Landasan Teori

Berikut ini merupakan landasan teori yang berkaitan atau berhubungan dengan penelitian ini.

a. Posyandu

Posyandu adalah singkatan dari Pos Pelayanan Terpadu, yang ialah unit pelayanan kesehatan dasar yang menyelenggarakan berbagai kegiatan untuk memantau, melindungi, dan meningkatkan kesehatan ibu, anak, dan keluarga[10]. Konsep Posyandu dikembangkan berdasarkan prinsip keterpaduan, partisipasi masyarakat, preventif, promotif, dan kuratif dalam pelayanan kesehatan[11].

Posyandu memiliki peran penting dalam kesehatan masyarakat, terutama dalam upaya meningkatkan kualitas hidup dan kesehatan ibu, anak, dan keluarga. Beberapa peran utama Posyandu meliputi:

- 1) Pemantauan tumbuh kembang anak balita: Posyandu melakukan pemantauan rutin terhadap pertumbuhan fisik, perkembangan motorik, dan perkembangan kognitif anak balita untuk mengidentifikasi potensi gangguan tumbuh kembang[12].
- 2) Imunisasi: Posyandu menjadi tempat pemberian imunisasi bagi bayi dan anak-anak, sehingga dapat melindungi mereka dari penyakit-penyakit infeksi[13].
- 3) Pelayanan kesehatan ibu: Posyandu juga memberikan pelayanan kesehatan kepada ibu hamil, seperti pemeriksaan kehamilan, persalinan, dan perawatan pasca melahirkan[14].

b. *Stunting*

Stunting merupakan kondisi gagal tumbuhnya perkembangan tubuh pada anak balita yang ditandai dengan pertumbuhan fisik yang terhambat, keterlambatan perkembangan fisik dan kognitif, serta tinggi badan yang lebih pendek dari standar usia atau tinggi badan yang seharusnya dicapai[15]. *Stunting* pada anak balita umumnya terjadi pada periode 0-5 tahun dan dapat memiliki dampak jangka panjang terhadap kesehatan, perkembangan, dan produktivitas anak di kemudian hari[16].

Stunting pada anak balita memiliki dampak jangka panjang yang signifikan, antara lain:

- 1) Gangguan perkembangan dan pertumbuhan: Anak dengan *stunting* cenderung memiliki pertumbuhan yang terhambat dan keterlambatan perkembangan fisik, kognitif, dan motorik[17].
- 2) Risiko penyakit: Anak balita yang mengidap *stunting* lebih rentan terhadap penyakit infeksi, gangguan sistem kekebalan tubuh, dan penyakit kronis di kemudian hari[18].
- 3) Gangguan kognitif: *Stunting* dapat mempengaruhi perkembangan kognitif anak, termasuk kemampuan belajar, memori, dan keterampilan kognitif lainnya[19].

Menurut peraturan Menteri Kesehatan Republik Indonesia nomor 2 tahun 2020, penilaian kategori dan ambang batas status gizi anak tersaji pada tabel berikut[20]:

Tabel 2.2 Kategori dan ambang batas status gizi anak.

Indeks	Kategori Status Gizi	Ambang Batas (Z-Score)
Panjang Badan atau Tinggi Badan menurut Umur (PB/U atau TB/U) anak usia 0 - 60 bulan	Sangat pendek (<i>severely stunted</i>)	< -3 SD
	Pendek (<i>stunted</i>)	- 3 SD sd <- 2 SD
	Normal	-2 SD sd +3 SD
	Tinggi	> +3 SD

Rumus perhitungan *Z-Score* TB/U atau PB/U dapat dihitung sebagai berikut:

$$Z \text{ Score} = \frac{TB \text{ anak} - TB \text{ Median acuan}}{\text{Nilai simpang baku rujukan}} \quad (1)$$

Dimana terdapat dua kategori dalam menghitung *Z-score*, yaitu:

- Bila nilai “TB anak” hasil pengukuran \geq nilai “TB Median Acuan”, maka rumusnya:

$$Z \text{ Score} = \frac{TB \text{ anak} - TB \text{ Median acuan}}{SD \text{ Upper}} \quad (2)$$

$$Z \text{ Score} = \frac{TB \text{ anak} - TB \text{ Median acuan}}{+1SD - \text{Median}} \quad (3)$$

- Bila nilai “TB anak” hasil pengukuran \leq nilai “TB Median Acuan”, maka rumusnya:

$$Z \text{ Score} = \frac{TB \text{ anak} - TB \text{ Median acuan}}{SD \text{ Lower}} \quad (4)$$

$$Z \text{ Score} = \frac{TB \text{ anak} - TB \text{ Median acuan}}{\text{Median} - -1SD} \quad (5)$$

Nilai standar baku antropometri dapat dilihat pada tabel berikut:

Tabel 2.3 Standar baku Panjang Badan menurut Umur (PB/U) anak laki-laki umur 0 – 60 bulan[20].

Umur (bulan)	Panjang Badan (cm)						
	-3 SD	-2 SD	-1 SD	<i>Median</i>	+1 SD	+2 SD	+3 SD
0	44.2	46.1	48.0	49.9	51.8	53.7	55.6
1	48.9	50.8	52.8	54.7	56.7	58.6	60.6
2	52.4	54.4	56.4	58.4	60.4	62.4	64.4
3	55.3	57.3	59.4	61.4	63.5	65.5	67.6
4	57.6	59.7	61.8	63.9	66.0	68.0	70.1
5	59.6	61.7	63.8	65.9	68.0	70.1	72.2
6	61.2	63.3	65.5	67.6	69.8	71.9	74.0
7	62.7	64.8	67.0	69.2	71.3	73.5	75.7
8	64.0	66.2	68.4	70.6	72.8	75.0	77.2
9	65.2	67.5	69.7	72.0	74.2	76.5	78.7
10	66.4	68.7	71.0	73.3	75.6	77.9	80.1
11	67.6	69.9	72.2	74.5	76.9	79.2	81.5
12	68.6	71.0	73.4	75.7	78.1	80.5	82.9
13	69.6	72.1	74.5	76.9	79.3	81.8	84.2
14	70.6	73.1	75.6	78.0	80.5	83.0	85.5
15	71.6	74.1	76.6	79.1	81.7	84.2	86.7
16	72.5	75.0	77.6	80.2	82.8	85.4	88.0
17	73.3	76.0	78.6	81.2	83.9	86.5	89.2
18	74.2	76.9	79.6	82.3	85.0	87.7	90.4
19	75.0	77.7	80.5	83.2	86.0	88.8	91.5

20	75.8	78.6	81.4	84.2	87.0	89.8	92.6
21	76.5	79.4	82.3	85.1	88.0	90.9	93.8
22	77.2	80.2	83.1	86.0	89.0	91.9	94.9
23	78.0	81.0	83.9	86.9	89.9	92.9	95.9
24 *	78.7	81.7	84.8	87.8	90.9	93.9	97.0
25	78.6	81.7	84.9	88.0	91.1	94.2	97.3
26	79.3	82.5	85.6	88.8	92.0	95.2	98.3
27	79.9	83.1	86.4	89.6	92.9	96.1	99.3
28	80.5	83.8	87.1	90.4	93.7	97.0	100.3
29	81.1	84.5	87.8	91.2	94.5	97.9	101.2
30	81.7	85.1	88.5	91.9	95.3	98.7	102.1
31	82.3	85.7	89.2	92.7	96.1	99.6	103.0
32	82.8	86.4	89.9	93.4	96.9	100.4	103.9
33	83.4	86.9	90.5	94.1	97.6	101.2	104.8
34	83.9	87.5	91.1	94.8	98.4	102.0	105.6
35	84.4	88.1	91.8	95.4	99.1	102.7	106.4
36	85.0	88.7	92.4	96.1	99.8	103.5	107.2
37	85.5	89.2	93.0	96.7	100.5	104.2	108.0
38	86.0	89.8	93.6	97.4	101.2	105.0	108.8
39	86.5	90.3	94.2	98.0	101.8	105.7	109.5
40	87.0	90.9	94.7	98.6	102.5	106.4	110.3
41	87.5	91.4	95.3	99.2	103.2	107.1	111.0
42	88.0	91.9	95.9	99.9	103.8	107.8	111.7
43	88.4	92.4	96.4	100.4	104.5	108.5	112.5

44	88.9	93.0	97.0	101.0	105.1	109.1	113.2
45	89.4	93.5	97.5	101.6	105.7	109.8	113.9
46	89.8	94.0	98.1	102.2	106.3	110.4	114.6
47	90.3	94.4	98.6	102.8	106.9	111.1	115.2
48	90.7	94.9	99.1	103.3	107.5	111.7	115.9
49	91.2	95.4	99.7	103.9	108.1	112.4	116.6
50	91.6	95.9	100.2	104.4	108.7	113.0	117.3
51	92.1	96.4	100.7	105.0	109.3	113.6	117.9
52	92.5	96.9	101.2	105.6	109.9	114.2	118.6
53	93.0	97.4	101.7	106.1	110.5	114.9	119.2
54	93.4	97.8	102.3	106.7	111.1	115.5	119.9
55	93.9	98.3	102.8	107.2	111.7	116.1	120.6
56	94.3	98.8	103.3	107.8	112.3	116.7	121.2
57	94.7	99.3	103.8	108.3	112.8	117.4	121.9
58	95.2	99.7	104.3	108.9	113.4	118.0	122.6
59	95.6	100.2	104.8	109.4	114.0	118.6	123.2
60	96.1	100.7	105.3	110.0	114.6	119.2	123.9

Tabel 2.4 Standar baku Panjang Badan menurut Umur (PB/U) anak perempuan umur 0 – 60 bulan[20].

Umur (bulan)	Panjang Badan (cm)						
	-3 SD	-2 SD	-1 SD	Median	+1 SD	+2 SD	+3 SD
0	43.6	45.4	47.3	49.1	51.0	52.9	54.7
1	47.8	49.8	51.7	53.7	55.6	57.6	59.5
2	51.0	53.0	55.0	57.1	59.1	61.1	63.2
3	53.5	55.6	57.7	59.8	61.9	64.0	66.1
4	55.6	57.8	59.9	62.1	64.3	66.4	68.6
5	57.4	59.6	61.8	64.0	66.2	68.5	70.7
6	58.9	61.2	63.5	65.7	68.0	70.3	72.5
7	60.3	62.7	65.0	67.3	69.6	71.9	74.2
8	61.7	64.0	66.4	68.7	71.1	73.5	75.8
9	62.9	65.3	67.7	70.1	72.6	75.0	77.4
10	64.1	66.5	69.0	71.5	73.9	76.4	78.9
11	65.2	67.7	70.3	72.8	75.3	77.8	80.3
12	66.3	68.9	71.4	74.0	76.6	79.2	81.7
13	67.3	70.0	72.6	75.2	77.8	80.5	83.1
14	68.3	71.0	73.7	76.4	79.1	81.7	84.4
15	69.3	72.0	74.8	77.5	80.2	83.0	85.7
16	70.2	73.0	75.8	78.6	81.4	84.2	87.0
17	71.1	74.0	76.8	79.7	82.5	85.4	88.2
18	72.0	74.9	77.8	80.7	83.6	86.5	89.4
19	72.8	75.8	78.8	81.7	84.7	87.6	90.6
20	73.7	76.7	79.7	82.7	85.7	88.7	91.7

21	74.5	77.5	80.6	83.7	86.7	89.8	92.9
22	75.2	78.4	81.5	84.6	87.7	90.8	94.0
23	76.0	79.2	82.3	85.5	88.7	91.9	95.0
24 *	76.7	80.0	83.2	86.4	89.6	92.9	96.1
25	76.8	80.0	83.3	86.6	89.9	93.1	96.4
26	77.5	80.8	84.1	87.4	90.8	94.1	97.4
27	78.1	81.5	84.9	88.3	91.7	95.0	98.4
28	78.8	82.2	85.7	89.1	92.5	96.0	99.4
29	79.5	82.9	86.4	89.9	93.4	96.9	100.3
30	80.1	83.6	87.1	90.7	94.2	97.7	101.3
31	80.7	84.3	87.9	91.4	95.0	98.6	102.2
32	81.3	84.9	88.6	92.2	95.8	99.4	103.1
33	81.9	85.6	89.3	92.9	96.6	100.3	103.9
34	82.5	86.2	89.9	93.6	97.4	101.1	104.8
35	83.1	86.8	90.6	94.4	98.1	101.9	105.6
36	83.6	87.4	91.2	95.1	98.9	102.7	106.5
37	84.2	88.0	91.9	95.7	99.6	103.4	107.3
38	84.7	88.6	92.5	96.4	100.3	104.2	108.1
39	85.3	89.2	93.1	97.1	101.0	105.0	108.9
40	85.8	89.8	93.8	97.7	101.7	105.7	109.7
41	86.3	90.4	94.4	98.4	102.4	106.4	110.5
42	86.8	90.9	95.0	99.0	103.1	107.2	111.2
43	87.4	91.5	95.6	99.7	103.8	107.9	112.0
44	87.9	92.0	96.2	100.3	104.5	108.6	112.7

45	88.4	92.5	96.7	100.9	105.1	109.3	113.5
46	88.9	93.1	97.3	101.5	105.8	110.0	114.2
47	89.3	93.6	97.9	102.1	106.4	110.7	114.9
48	89.8	94.1	98.4	102.7	107.0	111.3	115.7
49	90.3	94.6	99.0	103.3	107.7	112.0	116.4
50	90.7	95.1	99.5	103.9	108.3	112.7	117.1
51	91.2	95.6	100.1	104.5	108.9	113.3	117.7
52	91.7	96.1	100.6	105.0	109.5	114.0	118.4
53	92.1	96.6	101.1	105.6	110.1	114.6	119.1
54	92.6	97.1	101.6	106.2	110.7	115.2	119.8
55	93.0	97.6	102.2	106.7	111.3	115.9	120.4
56	93.4	98.1	102.7	107.3	111.9	116.5	121.1
57	93.9	98.5	103.2	107.8	112.5	117.1	121.8
58	94.3	99.0	103.7	108.4	113.0	117.7	122.4
59	94.7	99.5	104.2	108.9	113.6	118.3	123.1
60	95.2	99.9	104.7	109.4	114.2	118.9	123.7

c. *Machine learning*

Machine learning adalah bidang dalam ilmu komputer yang fokus pada pengembangan sistem dan algoritma yang dapat belajar secara otomatis dari data[21]. Dalam *machine learning*, data digunakan untuk melatih model, di mana model tersebut mempelajari pola dan hubungan dalam data tersebut[22]. Model ini kemudian dapat membuat prediksi atau mengambil keputusan berdasarkan data baru yang belum pernah dilihat sebelumnya[23]. Melalui konsep ini, *machine learning* memberikan kemampuan bagi komputer untuk belajar dan meningkatkan kinerjanya seiring dengan pengalaman.

Model *machine learning* dapat dilatih dengan menggunakan berbagai jenis algoritma, seperti regresi, pohon keputusan, atau jaringan saraf tiruan[24]. Proses pelatihan melibatkan pemberian model data yang telah dilabeli, di mana model secara iteratif mencari pola dan korelasi dalam data tersebut[25]. Tujuan utama dalam *machine learning* adalah untuk menciptakan model yang dapat menggeneralisasi dengan baik, yaitu mampu membuat prediksi yang akurat pada data baru yang belum pernah dilihat sebelumnya[26]. Sejalan dengan perkembangan metodologi dan kapasitas komputasional, teknik pembelajaran mesin kini telah diintegrasikan ke dalam beraneka ragam aplikasi, yang meliputi namun tidak terbatas pada pengenalan suara, identifikasi wajah, interpretasi tulisan tangan, deteksi objek, serta berbagai penerapan lainnya.

d. PHP

PHP, singkatan dari "PHP: *Hypertext Preprocessor*", merupakan bahasa pemrograman *scripting* yang secara khusus dikembangkan untuk pembuatan dan pengembangan aplikasi berbasis *web*. PHP memiliki sintaks yang mirip dengan bahasa pemrograman C dan menyediakan berbagai fitur yang memungkinkan pengembang untuk membangun aplikasi *web* yang dinamis dan interaktif[27].

Salah satu keunggulan PHP adalah kemampuannya untuk terintegrasi dengan HTML (*HyperText Markup Language*), sehingga memudahkan pengembang dalam menyusun halaman *web* yang dinamis. PHP dapat digunakan untuk mengolah formulir, berinteraksi dengan basis data, mengirim dan menerima *cookie*, mengelola sesi, dan banyak lagi. Selain itu, PHP memiliki banyak fungsi bawaan yang memudahkan dalam pengelolaan *string*, manipulasi *file*, pengolahan gambar, dan enkripsi data.

PHP juga didukung oleh berbagai *framework* populer seperti Laravel, CodeIgniter, dan Symfony, yang menyediakan struktur dan komponen tambahan untuk mempercepat pengembangan aplikasi *web*. Komunitas PHP yang besar dan aktif juga membuat tersedia berbagai sumber daya,

tutorial, dan dokumentasi yang membantu pengembang dalam mempelajari dan memperluas kemampuan PHP. Dengan semua kelebihan ini, PHP tetap menjadi salah satu pilihan utama dalam pengembangan aplikasi *web* yang handal dan efisien.

e. *Framework* Laravel

Framework Laravel dikembangkan oleh Taylor Otwell dan rilis pertama kali pada bulan Juni 2011. Laravel adalah kerangka kerja (*framework*) PHP yang populer untuk pengembangan aplikasi *web*. Laravel menyediakan seperangkat alat dan fitur yang kuat untuk mempermudah proses pengembangan, mempercepat produktivitas, dan memastikan keamanan aplikasi. *Framework* Laravel didesain berdasarkan arsitektur *Model-View-Controller* (MVC), yang secara struktural membedakan antara logika bisnis (*model*), antarmuka pengguna (*view*), dan mekanisme pengendalian (*controller*) dalam sebuah aplikasi. Laravel juga menyediakan fitur bawaan yang kuat seperti *authorization, database migration, eloquent ORM, authentication, notification & mail, validation, file storage, task scheduling, job queues, events & WebSocket, testing*, dan lain-lainnya[28].

f. MySQL

MySQL adalah sistem manajemen basis data relasional (RDBMS) yang sangat cepat, *multithread*, multi-pengguna, dan kuat. MySQL adalah perangkat lunak sumber terbuka dan dilisensikan di bawah Lisensi Publik Umum GNU (GPL). MySQL dirancang untuk menjadi sistem manajemen database yang cepat, andal, dan mudah digunakan. MySQL digunakan oleh jutaan orang di seluruh dunia untuk berbagai aplikasi, termasuk *web, e-commerce*, dan aplikasi *enterprise*[29].

g. Antropometri

Antropometri adalah cabang ilmu yang mempelajari pengukuran dan analisis dimensi fisik manusia. Istilah ini berasal dari bahasa Yunani, di mana "*anthropos*" berarti manusia dan "*metron*" berarti pengukuran. Antropometri berfokus pada pengukuran dan analisis berbagai dimensi dan proporsi tubuh manusia, termasuk tinggi badan, berat badan, lingkaran tubuh, lebar bahu, panjang lengan, dan sejumlah variabel lainnya[30].

Data antropometri diperoleh melalui pengukuran langsung individu atau sampel populasi menggunakan instrumen yang sesuai, seperti penggaris, jangka sorong, pita pengukur, atau peralatan lainnya yang dirancang khusus untuk tujuan pengukuran antropometri[31].

Data antropometri biasa digunakan untuk mengukur pertumbuhan dan perkembangan anak, menilai status gizi, dan mendeteksi masalah gizi seperti *stunting* (pertumbuhan terhambat) atau obesitas[32].

h. Python

Python adalah bahasa pemrograman tingkat tinggi yang terkenal karena sintaksis yang mudah dibaca dan dipahami. Dirancang untuk fokus pada kejelasan dan kekompakan kode, Python memungkinkan pengembang untuk mengekspresikan ide-ide kompleks dengan cara yang sederhana dan elegan. Python mendukung paradigma pemrograman yang beragam, termasuk pemrograman prosedural, pemrograman berorientasi objek, dan pemrograman fungsional[33].

Python juga dikenal dengan filosofi "*batteries included*", yang berarti bahwa bahasa ini menyediakan banyak fitur dan pustaka bawaan yang lengkap[34]. Selain itu, Python memiliki komunitas yang besar dan aktif yang mendukung pengembang dengan dokumentasi yang kaya, forum diskusi, dan sumber daya belajar yang melimpah.

Berikut ini adalah beberapa *library* yang sering digunakan oleh pengembang *machine learning* dalam Python :

a) NumPy

NumPy adalah *library* Python yang kuat untuk komputasi numerik. NumPy menyediakan struktur data N-dimensional *array* (*ndarray*) yang efisien dan beragam fungsi matematika yang dioptimalkan. Dengan NumPy, pengguna dapat melakukan operasi matematika kompleks secara efisien, seperti manipulasi array, perhitungan statistik, pengolahan sinyal, dan aljabar linear. NumPy juga memiliki integrasi yang kuat dengan pustaka-pustaka ilmiah lainnya, membuatnya menjadi fondasi yang penting dalam ekosistem pemrograman ilmiah Python.

b) Pandas

Pandas merupakan *library* yang digunakan secara luas dalam lingkungan Python untuk tujuan manipulasi serta analisis data. *Library* ini menawarkan struktur data yang dirancang untuk efisiensi serta fleksibilitas tinggi, khususnya struktur data *DataFrame*, yang memfasilitasi pengguna dalam menyimpan serta mengolah data dalam format tabel secara praktis. Dengan Pandas, pengguna dapat melakukan operasi seperti pemfilteran, penggabungan, pengindeksan, penghitungan statistik, dan transformasi data dengan mudah. Selain itu, Pandas juga menyediakan fungsi yang kuat untuk membaca dan menulis data dari berbagai format *file*, seperti CSV, Excel, SQL, dan banyak lagi. Dengan kecepatan, fleksibilitas, dan fungsionalitasnya, Pandas menjadi salah satu *library* yang paling banyak digunakan dalam analisis data dengan Python.

c) Scikit-learn

Scikit-learn merupakan *library* untuk pembelajaran mesin yang bersifat bebas biaya dan sumber terbuka, yang dikembangkan untuk bahasa pemrograman Python. Perpustakaan ini dikenal luas dan memiliki kemampuan yang beragam, sehingga banyak digunakan oleh komunitas pengembang pembelajaran mesin global. Scikit-learn menyajikan *array* algoritma pembelajaran mesin, yang mencakup

klasifikasi, regresi, dan klasterisasi. Selain itu, ia juga menyediakan serangkaian perangkat yang bermanfaat untuk persiapan data, evaluasi model, serta pelaksanaan prediksi.

d) Joblib

Joblib adalah *library* Python yang digunakan untuk menyimpan dan memuat objek Python dengan efisiensi tinggi. Joblib dirancang khusus untuk bekerja dengan tugas-tugas komputasi intensif, seperti pemrosesan data besar atau pelatihan model *machine learning* yang membutuhkan waktu lama. *Library* ini memungkinkan pengguna untuk menyimpan objek Python dalam format yang dioptimalkan untuk penyimpanan dan kemudian memuatnya kembali dengan cepat. Joblib juga menyediakan utilitas untuk membagi dan mendistribusikan tugas pemrosesan dalam lingkungan multiproses atau paralel. Dengan Joblib, pengguna dapat mempercepat proses pengolahan data dan pelatihan model yang kompleks, serta meningkatkan efisiensi penggunaan sumber daya komputasi.

e) Flask

Flask adalah *library* Python yang ringan dan fleksibel untuk membangun aplikasi *RestAPI web*. Flask memberikan kerangka kerja minimalis yang memudahkan pengembang dalam membuat aplikasi web dengan cepat dan efisien. Dengan menggunakan Flask, pengguna dapat dengan mudah mengatur *routing URL*, menangani permintaan HTTP, dan menghasilkan respon HTML atau JSON. *Library* ini juga menyediakan dukungan untuk manajemen sesi, dan pengembangan API. Dengan desain yang sederhana dan mudah dipahami, Flask menjadi pilihan yang populer untuk membangun aplikasi web ringan dengan Python.

i. *K-Nearest Neighbor* (KNN)

Algoritma klasifikasi *K-Nearest Neighbor* adalah suatu metode yang menetapkan klasifikasi sebuah entitas berdasarkan dominasi kategori dari

k tetangga terdekat. Proses ini melibatkan pengidentifikasian sekelompok objek dalam set data pelatihan yang memiliki jarak terdekat dengan objek yang sedang dianalisis dalam set data baru atau data uji[35].

Pada algoritma KNN, proses pelatihan hanya melibatkan penyimpanan data pelatihan yang dilabeli. Ketika ada data baru yang perlu diprediksi, algoritma mengukur jarak antara data tersebut dengan data pelatihan yang ada. Kemudian, KNN menggunakan keterangan dari K tetangga terdekat untuk mengklasifikasikan atau memprediksi data baru tersebut[36].

Salah satu aspek penting dalam algoritma KNN adalah pemilihan parameter K, yang menentukan jumlah tetangga terdekat yang akan digunakan untuk membuat prediksi. Jika K terlalu kecil, model dapat menjadi rentan terhadap *noise* dan *outlier*[37]. Sebaliknya, jika K terlalu besar, model dapat kehilangan kemampuan untuk menangkap pola yang lebih halus dalam data[38]. Oleh karena itu, pemilihan parameter K yang optimal menjadi perhatian penting dalam implementasi KNN. Algoritma *K-Nearest Neighbors* (KNN) dikenal karena kesederhanaan dan kejelasan konseptualnya, tetapi tidak luput dari beberapa limitasi, termasuk kepekaannya terhadap skala data dan potensi kebutuhan akan pemrosesan komputasional yang intensif ketika menghadapi volume data yang besar[39].

j. *Confusion Matrix*

Confusion matrix merupakan instrumen evaluatif yang sering diaplikasikan dalam pengujian model klasifikasi untuk menyajikan gambaran tentang efektivitas model tersebut. Matriks ini tersusun dalam bentuk tabel yang terdiri dari empat komponen utama: *true positive* (TP), *true negative* (TN), *false positive* (FP), dan *false negative* (FN)[40].

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Gambar 2.1 Tabel *confusion matrix* 2 kelas

Pada Gambar 2.1, terdapat ilustrasi tabel *confusion matrix* yang terdiri dari 2 kelas. Di dalam tabel ini, terdapat beberapa elemen yang dapat diukur nilainya, yang akan dijelaskan berikut ini[41]:

- True Positive* (TP) mengindikasikan jumlah kasus yang positif yang telah secara akurat terklasifikasi sebagai positif oleh model.
- True Negative* (TN) mengindikasikan jumlah kasus yang negatif yang dengan tepat teridentifikasi sebagai negatif oleh model.
- False Positive* (FP) mengindikasikan jumlah kasus yang negatif namun secara keliru dikategorikan sebagai positif.
- False Negative* (FN) mengindikasikan jumlah kasus yang positif tetapi secara salah terklasifikasi sebagai negatif.

Confusion matrix menyediakan suatu tinjauan menyeluruh terkait efektivitas model klasifikasi. Dari matriks tersebut, berbagai metrik evaluatif dapat diekstraksi dan dihitung, termasuk akurasi (*accuracy*), presisi (*precision*), sensitivitas (*recall*), serta skor F1 (*F1-score*). Berikut ini disampaikan elaborasi lebih terperinci mengenai keempat metrik tersebut[42]:

- Accuracy*

Akurasi(*Accuracy*) merupakan parameter yang mengukur tingkat kemampuan model dalam melakukan klasifikasi secara tepat.

Perhitungan ini dilakukan dengan cara membagi jumlah prediksi yang tepat (*true positive* dan *true negative*) dengan jumlah keseluruhan sampel yang ada. Parameter ini memberi pandangan umum mengenai efektivitas model dalam membuat prediksi yang akurat. Namun, akurasi bisa menghasilkan persepsi yang salah jika terjadi ketidakseimbangan dalam distribusi kelas di dalam dataset. Sebagai ilustrasi, dalam kondisi di mana terdapat 90% sampel negatif dan 10% sampel positif, sebuah model yang secara konstan memprediksi hasil negatif akan menunjukkan tingkat akurasi yang tinggi, meskipun sebenarnya model tersebut tidak efektif dalam mengenali sampel positif yang secara substansial mungkin lebih signifikan.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (6)$$

b) *Precision*

Precision adalah metrik yang mengevaluasi proporsi tingkat keakuratan dari hasil positif yang diidentifikasi oleh model. Perhitungan *precision* dilakukan dengan membagi jumlah *true positive* dengan total hasil positif yang diprediksi oleh model, yang meliputi *true positive* dan *false positive*. Indikator ini memberikan wawasan tentang seberapa efektif model dalam meminimalisasi jumlah prediksi positif yang tidak benar.

$$Precision = \frac{TP}{TP + FP} \quad (7)$$

c) *Recall*

Recall, juga dikenal sebagai Sensitivitas atau *True Positive Rate* (TPR), merupakan metrik yang mengevaluasi kapasitas model dalam mengenali secara valid hasil positif yang sebenarnya. Perhitungan dari *recall* dilakukan melalui pembagian antara jumlah hasil positif yang teridentifikasi dengan tepat (*true positive*) terhadap total hasil yang

dalam kenyataannya merupakan positif (agregasi dari *true positive* dan *false negative*). *Recall* memberikan persentase dari hasil positif yang sebenarnya yang berhasil diidentifikasi oleh model. *Recall* penting ketika meminimalkan hasil negatif palsu sangat penting atau ketika tidak ingin mengabaikan hasil positif yang sebenarnya. Contohnya, dalam deteksi kejahatan, *recall* yang tinggi penting untuk memastikan bahwa tidak ada kejahatan yang terlewatkan.

$$Recall = \frac{TP}{TP + FN} \quad (8)$$

d) *F1-score*

F1-score adalah ukuran gabungan dari *precision* dan *recall*, yang memberikan nilai tunggal yang memperhitungkan keduanya. *F1-score* dihitung sebagai *harmonic mean* dari *precision* dan *recall*. *Harmonic mean* memberi bobot yang lebih besar pada nilai terendah di antara dua metrik, sehingga *F1-score* akan lebih rendah jika salah satu metriknya rendah. *F1-score* berguna ketika kita ingin mencari keseimbangan antara *precision* dan *recall*. Jika keduanya memiliki nilai yang tinggi, *F1-score* juga akan tinggi. *F1-score* cocok untuk kasus di mana penting untuk mempertimbangkan secara seimbang antara keakuratan prediksi positif dan negatif.

$$F1\ Score = 2 \times \frac{precision \times recall}{precision + recall} \quad (9)$$

k. *Blackbox Testing*

Blackbox testing, juga dikenal sebagai pengujian fungsional, adalah metode pengujian perangkat lunak yang merujuk pada sebuah prosedur evaluasi perangkat lunak yang diimplementasikan tanpa mempertimbangkan struktur internal atau kode sumber dari aplikasi yang sedang dievaluasi. Pengujian jenis ini menitikberatkan pada spesifikasi fungsionalitas serta kebutuhan pengguna, dimana pengujian fokus pada

verifikasi *input* dan *output* yang diantisipasi dari sistem yang sedang diuji.[43].

Dalam *blackbox testing*, penguji tidak memiliki pengetahuan rinci tentang bagaimana sistem bekerja di dalamnya. Mereka menguji aplikasi seperti halnya pengguna akhir, menggunakan berbagai skenario dan *input* untuk mengamati respon sistem dan memverifikasi apakah *output* yang dihasilkan sesuai dengan yang diharapkan[44].

1. *Unified Modelling Language (UML)*

Unified Modeling Language (UML) merupakan suatu standar bahasa visualisasi yang diadopsi untuk pemodelan, perancangan arsitektur, serta pendokumentasian sistem perangkat lunak. UML menyediakan notasi grafis yang komprehensif dan multifaset untuk merepresentasikan elemen-elemen berbeda dari sebuah sistem, yang mencakup struktur, interaksi antar komponen, serta dinamika perilakunya. Bahasa ini digunakan oleh para pengembang perangkat lunak untuk berkomunikasi, berkolaborasi, dan memahami kompleksitas sistem yang sedang dibangun.

UML memberikan kerangka kerja yang kaya dan sistematis untuk menggambarkan berbagai aspek perangkat lunak, memfasilitasi analisis, desain, dan dokumentasi sistem secara efisien. Dengan notasi grafis yang konsisten dan standar, UML memungkinkan tim pengembangan untuk berkomunikasi secara lebih efektif dan membangun pemahaman yang sama tentang sistem yang sedang dikembangkan.

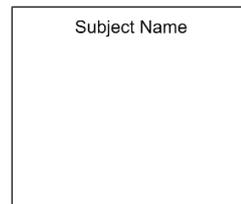
Dalam penelitian ini mengadopsi penggunaan diagram-diagram *Unified Modeling Language (UML)* yang berorientasi pada kategori perilaku, yang meliputi diagram *use case*, *activity diagram*, dan *sequence diagram*. Berikut diagram-diagram *Unified Modeling Language (UML)* yang akan digunakan, yaitu:

a) *Use Case Diagram*

Use Case Diagram adalah jenis diagram dalam *Unified Modeling Language (UML)* yang digunakan untuk menggambarkan interaksi

antara aktor eksternal dan fungsionalitas sistem yang diidentifikasi sebagai *use case* (skenario penggunaan).

- 1) *Use Case*: Mewakili entitas eksternal yang berinteraksi dengan sistem, seperti pengguna, sistem lain, atau perangkat eksternal.



Gambar 2.2 Notasi *use case*

- 2) *Actor*: Mewakili entitas eksternal yang berinteraksi dengan sistem, seperti pengguna, sistem lain, atau perangkat eksternal.



Gambar 2.3 Notasi *actor*

- 3) *Association Relationship*: Menunjukkan hubungan antara aktor dan use case, menunjukkan keterlibatan aktor dalam suatu skenario penggunaan.



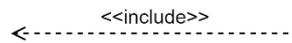
Gambar 2.4 Notasi *association relationship*

- 4) *Generalization Relationship*: Menunjukkan hubungan hierarkis antara *use case*, dengan *use case* induk berperan sebagai *use case* umum dan *use case* anak mewarisi fungsionalitas dari *use case* induk.



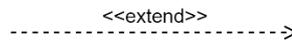
Gambar 2.5 Notasi *generalization relationship*

- 5) *Include Relationship*: Menunjukkan bahwa satu *use case* melibatkan atau memerlukan *use case* lain dalam eksekusinya.



Gambar 2.6 Notasi *include relationship*

- 6) *Exclude Relationship*: Menunjukkan bahwa suatu *use case* dapat memperluas atau menggantikan bagian dari *use case* lain dalam eksekusinya.

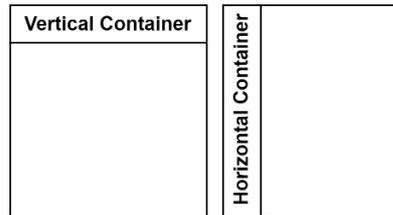


Gambar 2.7 Notasi *exclude relationship*

b) *Activity Diagram*

Activity Diagram merupakan salah satu tipe diagram yang terdapat dalam *Unified Modeling Language* (UML) yang bertujuan untuk mengilustrasikan aliran prosedural kerja atau rangkaian operasional aktivitas yang terjadi dalam suatu proses atau fungsi khusus pada sistem perangkat lunak. *Activity Diagram* terdiri dari beberapa elemen, yaitu :

- 1) *Swimline* adalah elemen yang menggambarkan wadah dari setiap aktivitas yang terjadi.



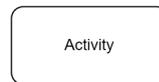
Gambar 2.8 Notasi *swimline*

- 2) *Initial node* adalah elemen yang menandakan status awal dari aktivitas yang dilakukan oleh sistem.



Gambar 2.9 Notasi *initial node*

- 3) *Activity* adalah elemen yang menandakan aktivitas pada sistem, biasanya aktivitas di dalam elemen ini diawali dengan kata kerja.



Gambar 2.10 Notasi *activity*

- 4) *Decision* adalah menunjukkan cabang pengambilan keputusan dalam alur aktivitas, di mana kondisi tertentu harus dipenuhi sebelum aktivitas berikutnya dapat dieksekusi.



Gambar 2.11 Notasi *decision*

- 5) *Join* digunakan untuk menggabungkan kembali jalur-jalur tersebut menjadi satu alur tunggal.



Gambar 2. 12 Notasi *join*

- 6) *Control Flow* adalah elemen yang menunjukkan aliran alur dari satu aktivitas ke aktivitas lain, menggambarkan urutan dan ketergantungan antara aktivitas.



Gambar 2.13 Notasi *control flow*

- 7) *Final Node* adalah elemen yang menunjukkan titik akhir atau terminasi dari *activity Diagram*.

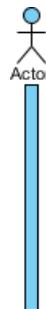


Gambar 2.14 Notasi *final node*

c) *Sequence Diagram*

Sequence Diagram merupakan varian diagram yang termasuk dalam kerangka *Unified Modeling Language (UML)*, yang dirancang untuk memvisualisasikan urutan interaksi antarobjek dalam konteks sistem perangkat lunak, termasuk ilustrasi pertukaran pesan yang terjadi antar objek-objek tersebut. *Sequence Diagram* terdiri dari beberapa elemen, yaitu :

- 1) *Actor*: jenis peran yang diperankan oleh suatu entitas yang berinteraksi dengan subjek.



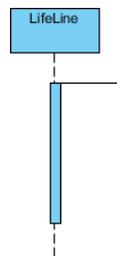
Gambar 2.15 Notasi *actor*

- 2) *Lifeline*: mewakili *individual participant* dalam interaksi.



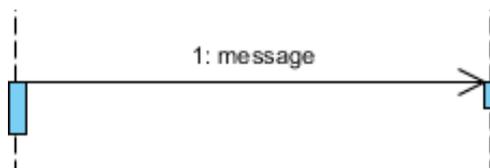
Gambar 2.16 Notasi *lifeline*

- 3) *Activations*: Persegi panjang tipis pada *lifeline* mewakili periode selama elemen melakukan operasi.



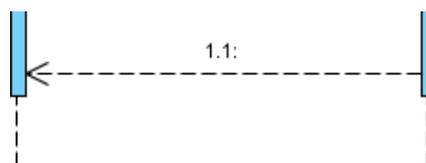
Gambar 2.17 Notasi *activations*

- 4) *Call message*: merupakan suatu tipe pesan yang menyimbolkan solisitasi untuk pelaksanaan operasi oleh *lifeline* yang menjadi target.



Gambar 2.18 Notasi *call message*

- 5) *Return message*: merupakan kategori pesan yang bertujuan untuk menyalurkan informasi respons ke *call message* yang telah ditransmisikan sebelumnya dan yang memiliki hubungan korespondensi dengan pesan tersebut.



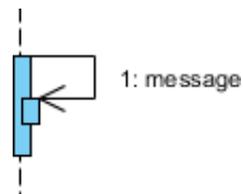
Gambar 2.19 Notasi *return message*

- 6) *Self message*: Call message: adalah jenis pesan yang mewakili permintaan operasi lifeline target.



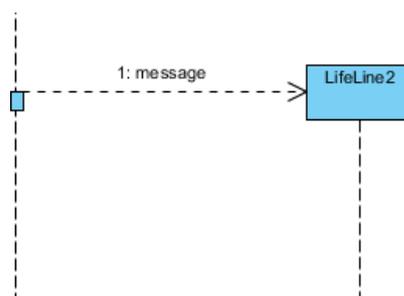
Gambar 2.20 Notasi *self message*

- 7) *Recursive message*: merupakan tipe pesan yang mengindikasikan pemanggilan sebuah pesan dari *lifeline* yang identik. Sasaran dari pesan ini mengacu pada aktivasi yang berlokasi di atas aktivasi di mana pesan tersebut diinisiasi.



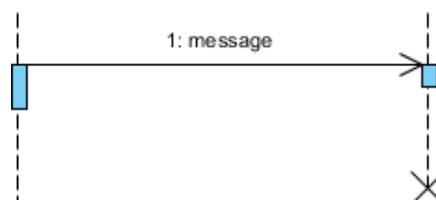
Gambar 2.21 Notasi *recursive message*

- 8) *Create message*: adalah jenis pesan yang merepresentasikan instantiasi dari (target) *lifeline*.



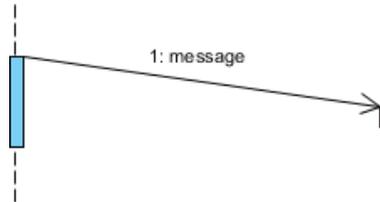
Gambar 2.22 Notasi *create message*

- 9) *Destroy message*: adalah jenis pesan yang mewakili permintaan untuk menghancurkan target *lifeline*.



Gambar 2.23 Notasi *destroy message*

10) *Duration message*: menunjukkan jarak antara dua instan waktu untuk pemanggilan pesan.



Gambar 2.24 Notasi *duration message*

11) *Note* (komentar): memberikan kemampuan untuk mengaitkan catatan tambahan pada elemen tertentu. Catatan tersebut tidak mengubah makna semantik dari elemen, namun dapat memuat informasi yang memberikan nilai tambah bagi individu yang melakukan pemodelan.

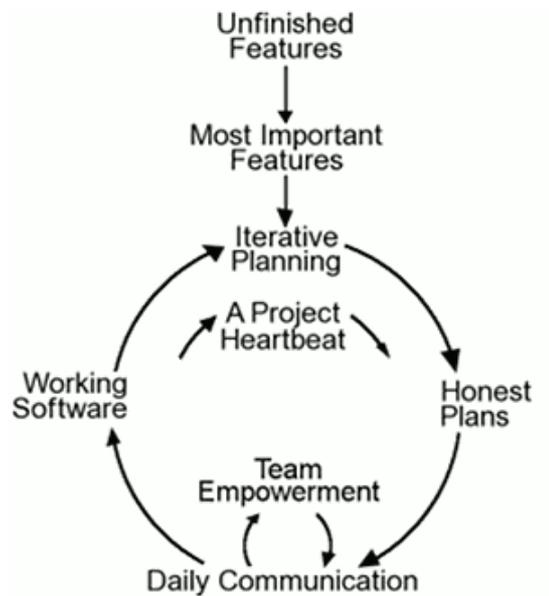


Gambar 2.25 Notasi *note*

m. Metode Pengembangan *Agile*

Metode pengembangan *Agile* adalah pendekatan kolaboratif dalam pengembangan perangkat lunak yang menekankan adaptabilitas, fleksibilitas, dan iterasi. Metode ini berfokus pada pengiriman nilai bisnis yang cepat dan responsif terhadap perubahan. *Agile* menggunakan pendekatan iteratif dan inkremental, di mana pengembangan dilakukan dalam siklus pendek yang disebut *sprint*. Setiap *sprint* memiliki tujuan yang jelas dan *deliverables* yang dihasilkan dalam waktu singkat, biasanya 1 hingga 4 minggu[45].

n. *Extreme Programming*



Gambar 2.26 Diagram alur XP.

Extreme Programming (XP) adalah metode pengembangan perangkat lunak yang tergolong dalam pendekatan *Agile*. XP menekankan pada adaptabilitas tim pengembangan. Metode ini menawarkan pendekatan yang sangat fleksibel dan berfokus pada pengiriman perangkat lunak yang berkualitas tinggi dalam waktu yang cepat[46].

XP memiliki beberapa praktik inti yang meliputi pengiriman perangkat lunak secara teratur dalam iterasi pendek, dan pemisahan pekerjaan kecil yang dapat diselesaikan dalam waktu singkat.