

BAB 2
LANDASAN TEORI

2.1. Perbandingan dengan Penelitian Terdahulu

Beberapa penelitian terkait dengan penelitian pada tugas akhir ini dapat dilihat pada Tabel 2.1.

Tabel 2.1. Perbandingan dengan Penelitian Terdahulu

No	Peneliti	Judul	Persamaan	Perbedaan
1	(Shofia dkk., 2017)	Sistem Pakar Diagnosis Penyakit Demam: DBD, Malaria dan <i>Tifoid</i> menggunakan Metode <i>K-Nearest Neighbor – Certainty Factor</i>	<i>Input:</i> 14 Gejala Demam <i>Tifoid</i> , DBD, dan Malaria yang sama <i>Output:</i> Penyakit Demam <i>Tifoid</i> , DBD, dan Malaria	<i>Input:</i> Shofia, dkk: 15 gejala penyakit Demam <i>Tifoid</i> , DBD, dan Malaria sebagai parameter Surayya: 29 gejala penyakit Demam <i>Tifoid</i> , DBD, dan Malaria sebagai parameter Algoritma: Shofia, dkk: Metode <i>K-Nearest Neighbor-Certainty Factor</i> Surayya: Metode NWKNN
2	(Rosyidah dkk., 2019)	Diagnosis Hama Penyakit Tanaman Bawang Merah Menggunakan Metode <i>Neighbor Weighted K-</i>	Algoritma: Metode yang digunakan adalah NWKNN	Penerapan: Rosyidah, dkk: Diagnosa hama dan penyakit tanaman Bawang Merah

Tabel 2.1. (Lanjutan)

No	Peneliti	Judul	Persamaan	Perbedaan
		<i>Nearest Neighbor</i> (NWKNN)		berdasarkan gejala Surayya: Klasifikasi penyakit Demam <i>Tifoid</i> , DBD, dan Malaria berdasarkan gejala.
3	(Maliha dkk., 2019)	Penerapan metode <i>Neighbor Weighted K-Nearest Neighbor</i> dalam Klasifikasi <i>Diabetes Mellitus</i>	Algoritma: Metode yang digunakan adalah NWKNN	Penerapan: Maliha, dkk: Klasifikasi penyakit <i>Diabetes Mellitus</i> Tipe 1 dan Tipe 2 berdasarkan gejala. Surayya: Klasifikasi penyakit Demam <i>Tifoid</i> , DBD, dan Malaria berdasarkan gejala.

2.2. Data Mining

Data *mining* adalah proses penggalian informasi dan pola yang bermanfaat dari sekumpulan data yang sangat besar. Tujuan data *mining* adalah untuk menemukan pola yang sebelumnya tidak diketahui. Data *mining* terdiri dari pengumpulan data, ekstraksi data, analisis data, dan statistik data. (Arhami & Nasir, 2020). Ada beberapa tugas yang dapat dilakukan oleh data *mining* dalam proses pemecahan masalah, yaitu (Larose, 2005):

1. Deskripsi (*Description*)

Deskripsi merupakan metode untuk menggambarkan kemungkinan penjelasan dari suatu pola dan tren yang ada di dalam data.

2. Estimasi (*Estimation*)

Estimasi merupakan metode untuk memperkirakan sesuatu hal yang belum pernah ada sebelumnya. Metode ini mirip dengan klasifikasi, perbedaannya hanya pada variabel target yang dihasilkan. Pada metode estimasi variabel target yang dihasilkan berbentuk numerik (angka) bukan kategori. Contoh algoritma pada metode estimasi adalah Regresi Linier, Confidence Interval Estimations, dan lain sebagainya.

3. Prediksi (*Prediction*)

Prediksi merupakan metode untuk memperkirakan atau meramalkan suatu kejadian di masa yang akan datang atau belum pernah terjadi. Metode ini mirip dengan klasifikasi dan estimasi. Contoh algoritma pada metode prediksi adalah *Decision Tree*, *Neural Network*, *K-Nearest Neighbor (KNN)*, dan lain sebagainya.

4. Klasifikasi (*Classification*)

Klasifikasi merupakan metode untuk mengelompokkan sesuatu berdasarkan kategori atau kelas. Pada metode klasifikasi memiliki variabel target berbentuk kategori. Contoh algoritma pada metode klasifikasi adalah *K-Nearest Neighbor*, *Neural Network*, *Decision Tree*, dan lain sebagainya.

5. Klustering (*Clustering*)

Klustering merupakan metode untuk pengelompokan sebuah objek ke dalam kelas objek yang serupa. Metode klustering berbeda dengan klasifikasi karena tidak mempunyai variabel target untuk mendapatkan hasil pengelompokan. Contoh algoritma pada metode klustering adalah *K-Means*, *K-Medoids*, dan lain sebagainya.

6. Asosiasi (*Association*)

Asosiasi merupakan metode untuk menemukan aturan suatu kombinasi atribut atau biasanya dikenal di dalam dunia bisnis sebagai analisis keranjang pasar dengan mengidentifikasi produk-produk yang seringkali secara bersamaan dibeli oleh pembeli. Contoh algoritma pada metode asosiasi adalah Apriori dan Algoritma GRI.

2.3. Neighbor Weighted K-Nearest Neighbor (NWKNN)

Metode *Neighbor Weighted K-Nearest Neighbor* (NWKNN) merupakan modifikasi dari metode KNN untuk menyelesaikan masalah data yang tidak seimbang (Tan, 2005). Pada metode NWKNN dilakukan pembobotan pada kategori atau target kelas yang akan digunakan dalam perhitungan skor untuk mendapatkan hasil akhir dari klasifikasi. Kategori kelas dengan jumlah data terkecil akan diberikan nilai bobot yang besar dan kategori kelas dengan jumlah data terbesar akan diberikan nilai bobot yang kecil. Langkah-Langkah pada algoritma NWKNN adalah sebagai berikut (Fadila dkk., 2016):

1. Menentukan nilai ketetanggaan (K) dan nilai *exp* (E).
2. Menghitung nilai kedekatan ketetanggaan antara data uji dan data latih (menggunakan *Euclidean Distance* atau *Cosine similarity / CosSim*). Untuk menghitung nilai kedekatan ketetanggaan menggunakan rumus *Cosine Similarity (CosSim)* dapat dilihat pada Persamaan 2.1.

$$\text{CosSim}(q, d_j) = \frac{\vec{a}_j \cdot \vec{q}}{|\vec{a}_j| \cdot |\vec{q}|} = \frac{\sum_{i=1}^m (w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^m (w_{ij})^2 \cdot \sum_{i=1}^m (w_{iq})^2}} \quad (2.1)$$

Keterangan:

- q : Data Uji
d_j : Data Latih
W_{ij} : Bobot nilai i pada data latih j
W_{iq} : Bobot nilai i pada data uji q
m : Banyak jumlah data

3. Mengurutkan hasil perhitungan nilai kedekatan ketetanggaan berdasarkan jarak atau kedekatan kelompok/*similarity* dengan cara mengurutkan dari nilai yang tertinggi ke nilai yang terendah.
4. Mengumpulkan jumlah data pada masing-masing kategori klasifikasi.
5. Dilakukan perhitungan bobot dengan Persamaan 2.2 (Tan, 2005).

$$\text{Weight}_i = \frac{1}{\left(\frac{\text{Num}(C_i^d)}{\text{Min}\{\text{Num}(C_j^d) | j=1, \dots, k^*\}} \right)^{1/E}} \quad (2.2)$$

Keterangan:

- Weight_i* : Bobot kelas i

$Num(C_i^d)$: Banyaknya data latih d pada kelas i

$Num(C_j^d)$: Banyaknya data latih d pada kelas j , dimana j termasuk dalam himpunan k tetangga terdekat

Exp : Eksponen (nilai E lebih dari 1)

6. Setelah mendapat nilai bobot dilanjutkan dengan menghitung nilai skor dengan cara mengalikan setiap jumlah hasil nilai kedekatan ketetanggan dari tiap kelas berdasarkan nilai K dengan masing-masing bobot, perhitungan nilai skor dapat menggunakan Persamaan 2.3 (Tan, 2005).

$$Score(X, C_i) = Weight_i * \left(\sum_{d \in d_{jNWKNN}(x)} \left(\left(Sim(d, d_j) \right) * \delta(d_j, C_i) \right) \right) \quad (2.3)$$

Keterangan:

$Weight_i$: Bobot kelas i

$d_{jNWKNN}(x)$: Data latih d_j pada kumpulan tetangga terdekat data uji X

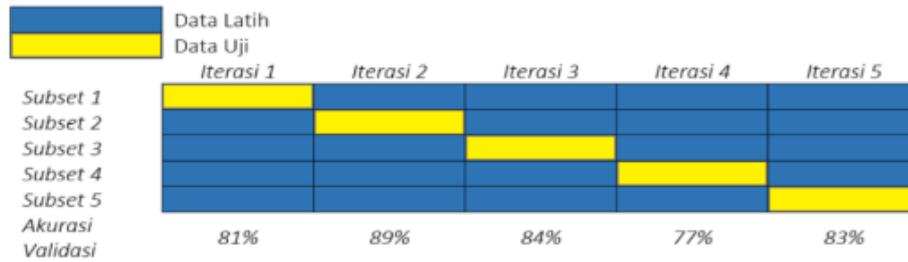
$\delta(d_j, C_i)$: Akan bernilai 1 jika nilai jarak $\in C_i$ dan bernilai 0 jika nilai jarak $\notin C_i$

$Sim(d, d_j)$: Nilai CosSim antara data uji dan data latih

C_i : Jenis atau kelas i

2.4. *K-Fold Cross Validation*

Metode *K-Fold Cross Validation* merupakan salah satu metode *Cross Validation*. Konsep *K-Fold Cross Validation* yaitu dengan membagi dataset ke dalam subset terpisah dengan ukuran yang sama sebanyak k atau *fold* (Mustika dkk., 2021). Menurut Kuhn dan Johnson (dalam Nisha Arya, 2022) dijelaskan bahwa nilai k yang digunakan biasanya adalah 5 atau 10, namun tidak ada aturan formal. Semakin besar nilai k , maka perbedaan jumlah antara data pelatihan dan data pengujian semakin kecil. Sehingga akan mengurangi kesalahan dalam prediksi dikarenakan kecilnya perbedaan antara nilai perkiraan dengan nilai sebenarnya. Kelebihan metode *K-Fold Cross Validation* yaitu mampu mengurangi waktu komputasi karena adanya proses iterasi dengan tetap menjaga keakuratan estimasi model (Mustika dkk., 2021). Pada gambar 2.1. menunjukkan bagaimana konsep *Cross Validation* bekerja dengan menggunakan $k = 5$ (Mustika dkk., 2021).



Gambar 2.1. Konsep Cross Validation

Sumber:(Mustika dkk., 2021)

Oleh karena itu, metode ini juga dapat menjamin bahwa setiap data telah digunakan untuk pelatihan dan pengujian. Terutama jika *dataset* berukuran besar, maka metode ini efisien untuk digunakan (Santoso dkk., 2020).

2.5. Confusion Matrix

Confusion matrix merupakan salah satu metode yang dapat digunakan untuk mengukur kinerja suatu model klasifikasi dalam memperoleh *accuracy*, *precision*, *recall*, dan *F1-Score*. Untuk mengukur kinerja suatu model klasifikasi dengan dimensi 3x3 dapat menggunakan tabel *confusion matrix multiclass* yang dapat dilihat pada Tabel 2.2.

Tabel 2.2. Tabel *Confusion Matrix Multiclass*

		<i>Predicted</i>		
		<i>Class 1</i>	<i>Class 2</i>	<i>Class 3</i>
<i>Actual</i>	<i>Class 1</i>	<i>True Positive (TP)</i>	C_{12}	C_{13}
	<i>Class 2</i>	C_{21}	<i>True Positive (TP)</i>	C_{23}
	<i>Class 3</i>	C_{31}	C_{32}	<i>True Positive (TP)</i>

Keterangan:

1. TP (*True Positive*) menunjukkan jumlah data hasil klasifikasi yang bernilai positif dan data aktualnya benar sebagai positif.
2. FP (*False Positive*) menunjukkan jumlah data hasil klasifikasi yang bernilai negatif tetapi data aktualnya sebagai positif. Contoh jika *class 1* sebagai True Positive, maka untuk nilai FP *class 1* = $C_{21} + C_{31}$.

3. TN (*True Negative*) menunjukkan jumlah data hasil klasifikasi yang bernilai negatif dan data aktualnya benar sebagai negatif. Contoh jika *class 1* sebagai True Positive, maka *class 2* dan *class 3* sebagai True Negative. Oleh karena itu, untuk nilai TN *class 1* = TP (*class 2*) + C₂₃ + C₃₂ + TP (*class 3*).
4. FN (*False Negative*) menunjukkan jumlah data hasil klasifikasi yang bernilai positif tetapi data aktualnya sebagai negatif. Contoh jika *class 1* sebagai true positive, maka *class 2* dan *class 3* sebagai True Negative. Oleh karena itu, untuk nilai FN *class 1* = C₁₂ + C₁₃

Accuracy (akurasi) didefinisikan sebagai pengukuran jumlah persentase terkoreksi benar oleh *classifier*. Berikut persamaan yang digunakan untuk perhitungan akurasi dapat dilihat pada Persamaan 2.4 (Setiawan, 2021).

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (2.4)$$

Precision (presisi) didefinisikan sebagai pengukuran hasil klasifikasi oleh sistem yang sesuai dengan kategori atau target aktualnya. Berikut persamaan yang digunakan untuk perhitungan presisi dapat dilihat pada Persamaan 2.5 (Setiawan, 2021).

$$Precision = \frac{TP}{TP+FP} \quad (2.5)$$

Recall didefinisikan sebagai tingkat keberhasilan sistem dalam mengenali sebuah kategori. Berikut persamaan yang digunakan untuk perhitungan *recall* dapat dilihat pada Persamaan 2.6 (Setiawan, 2021).

$$Recall = \frac{TP}{TP+FN} \quad (2.6)$$

F1-score didefinisikan sebagai kombinasi antara *precision* dan *recall*. Berikut persamaan yang digunakan untuk perhitungan *recall* dapat dilihat pada Persamaan 2.7 (Setiawan, 2021).

$$F1 - score = 2 * \frac{Precision \times Recall}{Precision + Recall} \quad (2.7)$$

2.6. MySQL

MySQL (*My Structure Query Language*) merupakan salah satu jenis dari perangkat lunak sistem manajemen basis data (DBMS) yang menggunakan bahasa

pemrograman *SQL. MySQL* bersifat *open source* karena termasuk kedalam DBMS yang multithread dan multi-user dibawah lisensi *GNU General Public License (GPL)* sehingga dapat digunakan secara gratis oleh semua orang (Anhar, 2010).

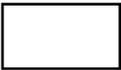
2.7. Flowchart

Flowchart merupakan penggambaran urutan logika yang terstruktur dari suatu prosedur pemecahan masalah yang dituliskan dengan simbol-simbol tertentu (Sitorus, 2015). *Flowchart* bertujuan untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai, dan mudah dipahami. Terdapat beberapa pedoman yang harus diikuti dalam pembuatan *flowchart*, yaitu (Hendraputra dkk., 2021):

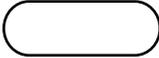
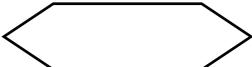
1. Penggambaran *flowchart* dimulai dari atas ke bawah dan dari kiri ke kanan.
2. Penggambaran tahapan di dalam *flowchart* harus ditunjukkan dengan jelas.
3. Kapan tahapan dimulai dan berakhir harus digambarkan dengan jelas.
4. Deskripsi tahapan di dalam *flowchart* digambarkan dengan menggunakan kata kerja.
5. Urutan tahapan di dalam *flowchart* harus dalam urutan yang benar.
6. Tahapan yang terpotong dan akan disambung di tempat lain harus digambarkan dengan jelas menggunakan simbol penghubung (konektor).
7. Penggambaran *flowchart* harus menggunakan simbol-simbol yang standar.

Simbol-simbol dalam *flowchart* memiliki bentuk dan fungsi yang berbeda-beda. Berikut simbol-simbol dalam *flowchart* yang sering digunakan dapat dilihat pada Tabel 2.3 (Sari, 2017).

Tabel 2.3. Simbol-simbol pada Flowchart

Simbol	Keterangan
 Proses	Menunjukkan suatu kegiatan atau proses yang dilakukan oleh komputer.
 Keputusan	Menunjukkan adanya keputusan atau kondisi tertentu dari sebuah proses atau tahapan.

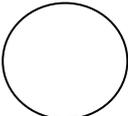
Tabel 2.3. (Lanjutan)

Simbol	Keterangan
 <i>Input / Output</i>	Merepresentasikan data masuk dan data keluar dari sebuah proses atau tahapan.
 Terminasi	Menunjukkan awal dan akhir dari sebuah proses atau tahapan
 Garis alir	Menunjukkan arah aliran suatu proses atau tahapan.
 <i>Preparation</i>	Menunjukkan proses inialisasi atau pemberian nilai awal pada variabel dan digunakan untuk perulangan for.
 Proses Terdefinisi	Menunjukkan rincian operasi yang berada di tempat lain

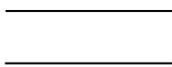
2.8. Data flow Diagram (DFD)

Data flow diagram (DFD) merupakan suatu diagram yang menggambarkan aliran data pada sistem menggunakan notasi-notasi dengan tujuan untuk memahami sistem secara logika, jelas, dan terstruktur (Ranatarisza & Noor, 2013). Berikut adalah simbol-simbol DFD menurut Yourdan dan De Marco dapat dilihat pada Tabel 2.4.

Tabel 2.4. Simbol-simbol DFD

Simbol	Keterangan
 Entitas luar	Entitas eksternal merupakan simbol yang digunakan untuk menggambarkan orang, organisasi, atau yang diluar sistem.
 Proses	Proses merupakan simbol yang digunakan untuk menggambarkan bagian dari sistem yang mentransformasikan data masukan menjadi data keluaran pada sistem.

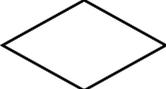
Tabel 2.4. (Lanjutan)

Simbol	Keterangan
 Data Store	Data <i>store</i> merupakan simbol yang digunakan untuk menggambarkan penyimpanan data seperti data disimpan dalam <i>database</i> .
 Alur Data	Alur data merupakan simbol yang menggambarkan aliran data yang mengalir antara entitas, proses, dan data <i>store</i> . Tujuannya utamanya adalah untuk menunjukkan arah dari satu sistem ke sistem yang lain.

2.9. Entity Relationship Diagram (ERD)

Entity relationship diagram (ERD) merupakan suatu diagram yang digunakan untuk menggambarkan relasi antar data secara konseptual. ERD berisikan kumpulan objek yang disebut entitas dan hubungan yang terjadi diantaranya yang disebut dengan relasi (*relationship*). Entitas memiliki atribut-atribut yang menjelaskan karakteristik entitas itu sendiri (Subakti dkk., 2022). Berikut adalah simbol-simbol ERD dapat dilihat pada Tabel 2.5 (Kusrini, 2007).

Tabel 2.5. Simbol-simbol ERD

Simbol	Keterangan
 Entitas	Entitas merupakan simbol yang digunakan untuk menggambarkan suatu objek atau benda secara unik, artinya dapat dibedakan dengan objek lainnya.
 Relasi	Relasi merupakan simbol yang digunakan untuk menggambarkan sebuah hubungan antara satu entitas dengan entitas lainnya.
 Atribut	Atribut merupakan simbol yang menggambarkan karakteristik dari suatu entitas. Atribut sebagai <i>field</i> atau kolom data yang digunakan dalam suatu entitas.

Tabel 2.5. (Lanjutan)

Simbol	Keterangan
<p style="text-align: center;">  Penghubung (<i>line</i>) </p>	Penghubung (<i>line</i>) merupakan simbol yang digunakan sebagai penghubung antara relasi dan entitas, dimana diantara kedua ujungnya memiliki kemungkinan jumlah pemakaian yang disebut dengan kardinalitas.

2.10. Python

Python merupakan bahasa pemrograman *interpreter* yang berarti sebuah kode akan langsung dieksekusi sesuai dengan instruksi yang ditulis dalam bahasa pemrograman tanpa mengubahnya menjadi *compiler*. Python memiliki struktur bahasa yang mudah dipelajari dan sederhana dibandingkan dengan bahasa pemrograman lainnya. Selain itu, python juga menyediakan banyak *library* yang sangat mudah untuk dipahami dan dapat digunakan untuk mendukung kebutuhan pada bidang keamanan jaringan, kecerdasan buatan, data *mining*, dan lain sebagainya (Rangkuti dkk., 2021).

2.11. Laravel

Laravel merupakan sebuah *framework* PHP yang dirancang untuk meningkatkan kualitas perangkat lunak guna mengurangi biaya pengembangan awal dan biaya pemeliharaan. Serta, menyediakan sintaks yang ekspresif, jelas, dan menghemat waktu. Laravel berada dibawah lisensi *Massachusetts Institute of Technology* (MIT), *open source*, dan dibangun menggunakan konsep MVC (Model View Controller) (Supardi & Sulaeman, 2019).

MVC merupakan sebuah pendekatan perangkat lunak yang memisahkan aplikasi berdasarkan komponen-komponen aplikasi seperti manipulasi data, *controller*, dan *user interface*. Terdapat komponen MVC dan pembagian tugasnya ialah sebagai berikut (Supardi & Sulaeman, 2019).

1. Model

Pada komponen ini, model merupakan bagian yang mewakili struktur data yang berguna dalam pengelolaan basis data seperti membaca data, memasukkan data ke dalam basis data, pembaruan data, dan menghubungkan data antar tabel.

2. *View*

Pada komponen ini, *view* merupakan bagian yang mengatur tampilan kepada *user* atau halaman *web*.

3. *Controller*

Pada komponen ini, *controller* merupakan bagian yang menjembatani model dan *view*. *Controller* berfungsi untuk menangani proses logika seperti mengambil permintaan, menginisialisasi, memanggil sebuah model, dan mengirimkan data yang akan ditampilkan pada *view*.

2.12. Demam

Demam pada umumnya ditandai dengan keadaan dimana suhu tubuh melebihi 37°C. Demam pada umumnya merupakan proses sistem kekebalan tubuh dalam melawan infeksi. Oleh karena itu, jika terserang demam tidak dianjurkan untuk langsung menurunkan suhu tubuh yang mana akan memperlambat ditemukannya penyebab dari demam tersebut dan dapat memperpanjang lama masa demam (Zein, 2012). Ada beberapa tipe demam yang tergantung pada penyebabnya (Zein, 2012) :

1. Demam Kontinua

Demam Kontinua ditandai dengan kondisi saat suhu tubuh tetap diatas normal sepanjang hari dan tidak ada fluktuasi suhu. Tipe demam ini biasanya disebabkan oleh infeksi saluran kemih, demam tifoid, brucellosis, dan pneumonia lobaris.

2. Demam Intermiten

Demam intermiten (demam putus-putus) ditandai dengan kondisi saat suhu tubuh mengalami kenaikan hanya dalam beberapa jam sehari dan kembali normal dalam beberapa jam atau adanya kondisi bebas periode demam. Tipe demam ini sering ditemukan pada penyakit Malaria dan sepsis.

3. Demam Remiten

Demam remiten ditandai dengan kondisi saat suhu tubuh mengalami kenaikan diatas normal sepanjang hari dan terjadi fluktuasi suhu. Tipe demam ini biasanya disebabkan oleh demam *tifoid* dan endokarditis.

2.12.1. Demam *Tifoid*

Demam *tifoid* merupakan penyakit infeksi akut saluran pencernaan yang disebabkan oleh bakteri *salmonella typhi* atau *salmonella paratyphi*. Gejala klinis yang sering terjadi pada demam tifoid diantaranya yaitu ditandai dengan demam, malaise, nyeri perut, dan konstipasi (Levani & Prasetya, 2020). Ada beberapa faktor yang mempengaruhi terjadinya penularan demam *tifoid* adalah sebagai berikut (Purnama, 2016):

1. Kurangnya kesadaran dalam menerapkan pola hidup sehat, seperti tidak membiasakan diri dalam menerapkan budaya cuci tangan di kehidupan sehari-hari.
2. Kurang terjaganya kebersihan pada makanan dan minuman, seperti makanan yang dicuci dengan air yang tercemar, makanan yang tercemar debu, sampah, dan dihindangi lalat, serta air minum yang tidak dimasak.
3. Sanitasi lingkungan yang buruk, dimana pengelolaan air limbah, kotoran, dan sampah yang buruk.
4. Tidak tersedianya air bersih untuk masyarakat tidak memadai.
5. Pasien atau *carrier tifoid* yang tidak diobati secara sempurna.
6. Belum membiasakan adanya program imunisasi untuk tifoid.

2.12.2. Demam Berdarah Dengue (DBD)

Demam Berdarah Dengue (DBD) merupakan penyakit infeksi yang disebabkan oleh virus *dengue* dari famili *flaviviridae*, genus *flavivirus* yang dapat hidup dan berkembang biak di dalam tubuh nyamuk dan manusia. Dengan kata lain, DBD dapat ditularkan melalui gigitan nyamuk yang terinfeksi virus *dengue* sehingga tubuh manusia yang terkena gigitan nyamuk akan terinfeksi virus dengue (Frida, 2019).

DBD biasanya ditandai dengan lama demam 2-7 hari yang disertai dengan gejala pendarahan, penurunan trombosit, dan menimbulkan kebocoran plasma. Serta dapat disertai dengan gejala yang tidak khas seperti nyeri kepala, nyeri otot dan tulang, ruam kulit atau nyeri mata. Penyebarluasan penularan penyakit DBD erat kaitannya dengan kepadatan dan mobilitas penduduk (Kementerian Kesehatan RI, 2017). Faktor-faktor lain yang mempengaruhi penyebarluasan DBD, yaitu (Kementerian Kesehatan RI, 2017):

1. Perilaku masyarakat
2. Perubahan iklim global
3. Pertumbuhan ekonomi
4. Ketersediaan air bersih

2.12.3. Malaria

Malaria merupakan penyakit infeksi yang disebabkan oleh parasit plasmodium yang ditularkan melalui gigitan nyamuk anopheles betina. Demam merupakan gejala utama pada Malaria. Pada periode awal sakit ditandai dengan kondisi tubuh menggigil dan diikuti dengan demam tinggi kemudian berkeringat. Selain gejala diatas, gejala lainnya pada penyakit Malaria yaitu nyeri kepala, mual, muntah, diare, nyeri otot dan sendi (Kementerian Kesehatan RI, 2020).

Gejala klinis pada Malaria sangat bervariasi dari ringan hingga berat yang dapat menyebabkan kematian. Gejala klinis Malaria juga sangat mirip dengan penyakit lain, seperti demam *tifoid*, demam berdarah *Dengue*, leptospirosis, chikungunya, dan lain sebagainya (Kementerian Kesehatan RI, 2020).