

BAB II

LANDASAN TEORI

2.1. Dehidrasi

2.1.1. Definisi

Dehidrasi merupakan kondisi kekurangan cairan tubuh karena jumlah cairan yang keluar lebih banyak daripada jumlah cairan yang masuk. Jika tubuh kehilangan banyak cairan, maka tubuh akan mengalami dehidrasi. Secara definisi, dehidrasi adalah suatu keadaan penurunan total air di dalam tubuh karena hilangnya cairan secara patologis, asupan air tidak adekuat (syarat), atau kombinasi keduanya. Dehidrasi terjadi karena pengeluaran air lebih banyak daripada jumlah yang masuk, dan kehilangan cairan ini juga disertai dengan hilangnya elektrolit.

Pada dehidrasi terjadi keseimbangan negatif cairan tubuh akibat penurunan asupan cairan dan meningkatnya jumlah air yang keluar (lewat ginjal, saluran cerna atau *insensible water loss/IWL*) atau karena adanya perpindahan cairan dalam tubuh. Berkurangnya volume total cairan tubuh menyebabkan penurunan volume cairan intrasel dan ekstrasel. Manifestasi klinis dehidrasi erat kaitannya dengan depleksi *volume* cairan intravaskuler. Proses dehidrasi yang berkelanjutan dapat menimbulkan syok hipovolemia yang akan menyebabkan gagal organ

dan kematian. (Dr. Purnamawati Sujud Pujiarto, Sp.A(K), 2015; Leksana, 2015)

2.2.2. Penyebab Dehidrasi

Mencari penyebab dehidrasi merupakan hal penting. Asupan cairan yang buruk, cairan keluar berlebihan, peningkatan *insensible water loss* (IWL), atau kombinasi hal tersebut dapat menjadi penyebab depleksi *volume* intravaskuler. Keberhasilan terapi membutuhkan identifikasi penyakit yang mendasari kondisi dehidrasi. Beberapa faktor patologis penyebab dehidrasi yang sering menurut (Leksana, 2015), yaitu:

- *Gastroenteritis Diarrhea* adalah etiologi paling sering.

Pada diare yang disertai muntah, dehidrasi akan semakin progresif. Dehidrasi karena diare menjadi penyebab utama kematian bayi dan anak di dunia.

- Stomatitis dan faringitis.

Rasa nyeri mulut dan tenggorokan dapat membatasi asupan makanan dan minuman lewat mulut.

- Ketoasidosis diabetes (KAD).

Disebabkan karena adanya diuresis osmotik dan berat badan turun akibat kehilangan cairan dan katabolisme jaringan.

- Demam

Demam dapat meningkatkan IWL dan menurunkan nafsu makan.

Selain hal di atas, dehidrasi juga dapat dicetuskan oleh kondisi *heat stroke*, tirotoksikosis, obstruksi saluran cerna, fibrosis sistik, diabetes insipidus, dan luka bakar.

2.2.3. Tipe Dehidrasi

Kehilangan cairan tubuh biasanya disertai gangguan keseimbangan elektrolit. Dehidrasi dapat dikategorikan berdasarkan osmolaritas dan derajat keparahannya. Kadar natrium serum merupakan penanda osmolaritas yang baik selama kadar gula darah normal. Berdasarkan perbandingan jumlah natrium dengan jumlah air yang hilang, dehidrasi dibedakan menjadi tiga tipe yaitu dehidrasi isotonik, dehidrasi hipertonik, dan dehidrasi hipotonik. Variasi kadar natrium mencerminkan jumlah cairan yang hilang dan memiliki efek patofisiologi berbeda.

- Dehidrasi isotonik (isonatremik).

Tipe ini merupakan yang paling sering (80%). Pada dehidrasi isotonik kehilangan air sebanding dengan jumlah natrium yang hilang, dan biasanya tidak mengakibatkan cairan ekstrasel berpindah ke dalam ruang intraseluler. Kadar natrium dalam darah pada dehidrasi tipe ini 135-145 mmol/L dan osmolaritas efektif serum 275-295 mOsm/L.

- Dehidrasi hipotonik (hiponatremik).

Natrium hilang yang lebih banyak daripada air. Penderita dehidrasi hipotonik ditandai dengan rendahnya kadar natrium serum

(kurang dari 135 mmol/L) dan osmolalitas efektif serum (kurang dari 270 mOsm/L). Karena kadar natrium rendah, cairan intravaskuler berpindah ke ruang ekstrasvaskuler, sehingga terjadi depleksi cairan intravaskuler. Hiponatremia berat dapat memicu kejang hebat; sedangkan koreksi cepat hiponatremia kronik (2 mEq/L/jam) terkait dengan kejadian mielinolisis pontin sentral.

- Dehidrasi hipertonik (hipernatremik).

Hilangnya air lebih banyak daripada natrium. Dehidrasi hipertonik ditandai dengan tingginya kadar natrium serum (lebih dari 145 mmol/L) dan peningkatan osmolalitas efektif serum (lebih dari 295 mOsm/L). Karena kadar natrium serum tinggi, terjadi pergeseran air dari ruang ekstrasvaskuler ke ruang intravaskuler. Untuk mengkompensasi, sel akan merangsang partikel aktif (idiogenik osmol) yang akan menarik air kembali ke sel dan mempertahankan volume cairan dalam sel. Saat terjadi rehidrasi cepat untuk mengoreksi kondisi hipernatremia, peningkatan aktivitas osmotik sel tersebut akan menyebabkan influks cairan berlebihan yang dapat menyebabkan pembengkakan dan ruptur sel; edema serebral adalah konsekuensi yang paling fatal. Dehidrasi secara perlahan dalam lebih dari 48 jam dapat meminimalkan risiko ini. (Leksana, 2015)

2.2.4. Derajat Dehidrasi

Derajat dehidrasi ditentukan berdasarkan keluhan dan gejala yang merefleksikan jumlah kekurangan cairan dan elektrolit yang terjadi. Cara terbaik untuk menentukan derajat dehidrasi adalah persentase kehilangan volume cairan yang bisa dihitung dari selisih berat badan sebelum sakit dan berat badan saat sakit dibagi dengan berat badan sebelum sakit. Namun data berat badan sebelum sakit masih sulit diperoleh terutama di negara-negara berkembang termasuk juga Indonesia. Melihat pentingnya penentuannya derajat dehidrasi dan sulitnya menghitung penurunan berat badan selama dehidrasi, maka *World Health Organization* (WHO) telah membuat penilaian derajat dehidrasi berdasarkan empat parameter penilaian gejala klinik, yaitu keadaan umum, mata, rasa haus, dan penilaian turgor kulit. Penilaian derajat dehidrasi WHO sangat mudah digunakan oleh masyarakat awam sehingga diharapkan dapat mengurangi komplikasi dehidrasi. Namun penilaian dehidrasi WHO ini pada umumnya bersifat subjektif. Sebagai akademisi dan pelayan kesehatan tentu perlu suatu penilaian dehidrasi dengan menggunakan kriteria objektif. (Dr. Purnamawati Sujud Pujiarto, Sp.A(K), 2015; Mupidah, Salekede, & Daud, 2014)

2.2.5. Klasifikasi Derajat Dehidrasi

Berikut klasifikasi derajat dehidrasi menurut (Leksana, 2015):

Tabel 2.1 Derajat Dehidrasi Berdasarkan Presentase Kehilangan Air

Sumber: (Leksana, 2015)

Derajat Dehidrasi	Dewasa	Bayi dan Anak
Dehidrasi Ringan	4% dari berat badan	5% dari berat badan
Dehidrasi Sedang	6% dari berat badan	10% dari berat badan
Dehidrasi Berat	8% dari berat badan	15% dari berat badan

Tabel 2.2 Derajat Dehidrasi Berdasarkan Skor WHO

Sumber: (Leksana, 2015)

Yang Dinilai	SKOR		
	A	B	C
Keadaan Umum	Baik	Lesu/haus	Gelisah, lemas, mengantuk hingga syok
Mata	Biasa	Cekung	Sangat cekung
Mulut	Biasa	Kering	Sangat kering
Turgor	Baik	Kurang	Jelek

Keterangan:

Skor: < 2 tanda di kolom B dan C : tanpa dehidrasi

> 2 tanda di kolom B : dehidrasi ringan – sedang

>= 2 tanda di kolom C : dehidrasi berat

Tabel 2.3 Tanda Klinis Dehidrasi

Sumber: (Leksana, 2015)

Deskripsi	Ringan	Sedang	Berat
Defisit Cairan	3-5%	6-8%	>10%
Hemodinamik	Takikardi Nadi lemah	Takikardi Nadi sangat lemah Volume kolaps Hipotensi ortostatik	Takikardi Nadi tak teraba Akral dingin, sianosis
Jaringan	Lidah kering Turgor turun	Lidah keriput Turgor kurang	Atonia Turgor buruk

Deskripsi	Ringan	Sedang	Berat
Urin	Pekat	Jumlah turun	Oliguria
SSP	Mengantuk	Apatis	Koma

2.2.6. Penanganan Kondisi Dehidrasi

Adanya beberapa kondisi dalam penanganan kondisi dehidrasi menurut (Leksana, 2015) yang dibagi menjadi dua tahap, yaitu:

- Tahap Pertama

Berfokus untuk mengatasi kedaruratan dehidrasi, yaitu syok hipovolemia yang membutuhkan penanganan cepat. Pada tahap ini dapat diberikan cairan kristaloid isotonik, seperti *ringer lactate* (RL) atau NaCL 0,9% sebesar 20 mL/kgBB. Perbaikan cairan intravaskuler dapat dilihat dari perbaikan takikardi, denyut nadi, produksi urin, dan status mental pasien. Apabila perbaikan belum terjadi setelah cairan diberikan dengan kecepatan hingga 60 mL/kgBB, maka etiologi lain syok harus dipikirkan (misalnya anafilaksis, sepsis, syok kardiogenik). Pengawasan hemodinamik dan golongan inotropik dapat diindikasikan.

- Tahap Kedua

Berfokus pada mengatasi defisit, pemberian cairan pemeliharaan dan penggantian kehilangan yang masih berlangsung. Kebutuhan cairan pemeliharaan diukur dari jumlah kehilangan cairan (urin, tinja) ditambah IWL. Jumlah IWL adalah antara 400-500 mL/m² luas permukaan tubuh dan dapat meningkat pada kondisi demam dan

takipnea. Berikut secara kasar kebutuhan cairan berdasarkan berat badan.

- Berat badan < 10 kg = 100 mL/kgBB
- Berat badan 10-20 kg = 1000 + 50 mL/kgBB untuk setiap kilogram berat badan di atas 10 kg
- Berat badan > 20 kg = 1500 + 20 mL/kgBB untuk setiap kilogram berat badan di atas 20 kg

2.2. Rapid Application Development (RAD)

Rapid Application Development (RAD) atau *rapid prototyping* adalah model proses pembangun perangkat lunak yang tergolong dalam teknik *incremental* (bertingkat). *Rapid Application Development (RAD)* menekankan pada siklus pembangunan pendek, singkat, dan cepat. Waktu yang singkat adalah batasan yang penting untuk model ini. *Rapid Application Development (RAD)* menggunakan metode iteratif (berulang) dalam mengembangkan sistem dimana *working model* (model kerja) sistem dikonstruksikan di awal tahap pengembangan dengan tujuan menetapkan kebutuhan (*requirement*) pengguna. Model kerja digunakan hanya sesekali sebagai basis desain dan implementasi sistem akhir. (Sagala, 2018)

Rapid Application Development (RAD) sendiri juga memiliki kelebihan dan kekurangan. Berikut kelebihan dari *Rapid Application Development (RAD)* menurut (Sagala, 2018), yaitu:

- a. Sangat berguna dilakukan pada kondisi *user* tidak memahami kebutuhan-kebutuhan apa saja yang digunakan pada proses pengembangan perangkat lunak.
- b. *Rapid Application Development* (RAD) mengikuti tahapan pengembangan sistem seperti umumnya, tetapi mempunyai kemampuan untuk menggunakan kembali komponen yang ada (*reuseable object*) sehingga pengembang tidak perlu membuat dari awal lagi dan waktu lebih singkat berkisar antara 60 hari sampai 90 hari.
- c. Dikarenakan mempunyai kemampuan untuk menggunakan komponen yang sudah ada dan waktu yang lebih singkat, maka membuat biaya menjadi lebih rendah dalam menggunakan *Rapid Application Development* (RAD).

Berikut kelemahan dari *Rapid Application Development* (RAD) menurut (Sagala, 2018), yaitu:

- a. Proyek yang besar dan berskala, *Rapid Application Development* (RAD) memerlukan sumber daya manusia yang memadai untuk menciptakan jumlah tim yang baik.
- b. *Rapid Application Development* (RAD) menuntut pengembang dan pelanggan memiliki komitmen dalam aktivitas *rapid fire* yang diperlukan untuk melengkapi sebuah sistem dalam waktu yang singkat. Jika komitmen tersebut tidak ada, maka proyek *Rapid Application Development* (RAD) akan gagal.

Kesimpulan setelah melihat kelebihan dan kekurangan dari metode *Rapid Application Development* (RAD) menurut (Sagala, 2018), yaitu:

- a. Apabila menggunakan metode *Rapid Application Development* (RAD) untuk mendapatkan suatu desain yang dapat diterima oleh konsumen dan dapat dikembangkan dengan mudah.
- b. Apabila menggunakan metode *Rapid Application Development* (RAD) untuk memberikan batasan-batasan pada suatu sistem supaya tidak mengalami perubahan.
- c. Apabila menggunakan metode *Rapid Application Development* (RAD) untuk menghemat waktu, dan kalau memungkinkan bisa menghemat biaya serta menghasilkan produk yang berkualitas.

Metode pengembangan *Rapid Application Development* (RAD) memiliki 4 tahapan siklus pengembangan menurut (Sagala, 2018), yaitu:

1. Fase Analisis Persyaratan

Fase ini memiliki tujuan untuk mengidentifikasi layanan, batasan, dan obyektifitas dari sistem pengumpulan data yang dilakukan terhadap *stakeholders*.

2. Fase Analisis *Modeling*

Fase ini memiliki tujuan untuk menganalisis semua kegiatan dalam arsitektur sistem secara keseluruhan dengan melibatkan identifikasi dan deskripsi abstraksi sistem perangkat lunak yang mendasar dan hubungan-hubungannya.

3. Fase Desain *Modeling*

Fase ini memiliki tujuan untuk melakukan perancangan sistem berdasarkan analisis yang telah dilakukan sebelumnya. Tahap analisis dan desain mengalami perulangan hingga diperoleh rancangan sistem yang benar-benar memenuhi kebutuhan.

4. Fase Konstruksi

Fase ini memiliki tujuan untuk menunjukkan *platform*, *hardware*, dan *software* yang digunakan serta batasan dalam implementasi, serta menguji performansi *prototype* perangkat lunak yang telah dibangun agar dapat diketahui apakah *prototype* tersebut telah sesuai dengan spesifikasi analisis dan perancangan yang telah diidentifikasi sebelumnya. Hasil akhir dari fase konstruksi adalah *platform*, *hardware*, dan *software* yang digunakan, serta daftar batasan implementasi, dan rencana pengujian.

2.3. *Waterfall*

Waterfall merupakan salah satu model pengembangan yang bersifat *linear* dari tahap awal pengembangan sistem, yaitu tahap perencanaan sampai tahap akhir pengembangan sistem, yaitu tahap pemeliharaan. *Waterfall* juga bersifat sistematis, dimana model ini berurutan dalam pembangunan *software*. Model pengembangan ini termasuk ke dalam model *generic* pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh Winston Royce pada tahun 1970, sehingga *waterfall* sering dianggap kuno, tetapi merupakan model yang paling

banyak dipakai dalam *Software Engineer* (SE). (Susanto & Andriana, 2016; Widiyanto, 2018)

Waterfall memiliki kelebihan dan kekurangan. Berikut kelebihan dari *waterfall* menurut (Widiyanto, 2018), yaitu:

- a. Tahapan prosesnya tidak akan berubah, dan teratur.
- b. Sangat baik digunakan dalam pembangunan *software* maupun program yang sudah jelas kebutuhannya.
- c. Memiliki kualitas yang baik.
- d. Dokumen pengembangan dengan sistem yang terorganisir, karena setiap fase harus terselesaikan sebelum melakukan ke fase selanjutnya.

Berikut kelemahan dari *waterfall* menurut (Widiyanto, 2018), yaitu:

- a. Sering terjadi perubahan saat berjalannya proyek, sehingga dapat menyebabkan masalah baru.
- b. Pengembangan proyek yang menjadi tahap-tahap fleksibel.
- c. Sulitnya saat mengalami perubahan saat proyek berlangsung.
- d. Proses yang harus dilakukan secara teratur karena dikerjakan secara bertahap.
- e. Perubahan ditengah pengerjaan proyek dapat menghambat proyek karena harus mengulang fase maupun tahapan proses yang dikerjakan.

- f. Karena model pengembangan *waterfall* ini dianjurkan untuk dikerjakan secara tim, maka adanya waktu kosong bagi beberapa anggota tim proyek karena harus menunggu anggota tim proyek lainnya menyelesaikan tugasnya.

Kesimpulan setelah melihat kelebihan dan kekurangan dari model pengembangan *waterfall* ini menurut (Widiyanto, 2018) adalah *waterfall* ini kurang baik bila proyek ingin dikerjakan dalam waktu singkat, dan terhambatnya proyek bila adanya perubahan di tengah-tengah pengerjaan proyek, maka dari itu harus dilakukannya dokumentasi saat proyek dikerjakan dengan model pengembangan ini.

2.4. Prototype

Prototype biasanya didefinisikan sebagai alat yang memberi ide bagi pembuat maupun *user* tentang cara sistem berfungsi dalam bentuk lengkapnya. Hasil dari *prototype* juga disebut dengan *prototyping*. *Prototyping* merupakan suatu teknik untuk mengumpulkan informasi tertentu mengenai kebutuhan-kebutuhan informasi penggunaan secara cepat. Berfokus pada penyajian dari aspek-aspek perangkat lunak tersebut yang akan nampak bagi pelanggan atau pemakai. (Susanto & Andriana, 2016; Widiyanto, 2018)

Prototype memiliki kelebihan dan kekurangan. Berikut kelebihan dari *prototype* menurut (Widiyanto, 2018), yaitu:

- a. Tim harus aktif dalam pengembangan sistem, sehingga hasil dari pengembangan akan menjadi mudah untuk disesuaikan dengan keinginan dan kebutuhan.
- b. Penentuan kebutuhan yang lebih mudah untuk dilakukan.
- c. Dapat mempersingkat waktu dalam pengembangan *software*.

Berikut kelemahan dari *prototype* menurut (Widiyanto, 2018), yaitu:

- a. Proses analisis dan perancangan yang terlalu singkat.
- b. Kurangnya fleksibel dalam menghadapi perubahan.
- c. Walaupun adanya perbaikan dari setiap versi *prototype*, tetapi *user* mungkin tidak menyadari jika versi dibuat tanpa memperhatikan kualitas untuk jangka panjang.
- d. Kurangnya kompromi implementasi dengan menggunakan sistem operasi yang tidak relevan dan algoritma yang tidak efisien.

Kesimpulan setelah melihat kelebihan dan kekurangan dari model pengembangan *prototype* ini menurut (Widiyanto, 2018) adalah kurang baik digunakan bila harus menyesuaikan dengan keinginan *user* misalnya dalam hal memberi contoh dari setiap modul yang dibuat untuk ditunjukkan kepada *user*. Jika modul sudah sesuai maka dapat melanjutkan ke tahap selanjutnya, tetapi

jika tidak atau belum sesuai, maka akan dilakukan perbaikan sesuai dengan permintaan *user*. Dalam pengembangan ini juga dibutuhkan tenaga ahli untuk membantu penaksiran risiko, karena hal ini juga pengembangan jadi membutuhkan waktu yang lama saat menerapkan metode ini sehingga dapat menghambat penelitian.

2.5. Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah sebuah kosakata umum berbasis objek dan diagram yang cukup efektif dalam memodelkan setiap proyek pengembangan sistem dari tahap analisis hingga tahap implementasinya. (Dennis, Wixom, & Roth, 2015)

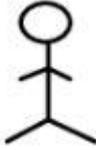
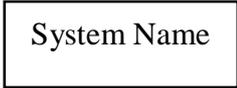
Berikut beberapa jenis *Unified Modeling Language* (UML) menurut (Dennis et al., 2015):

2.5.1. Use Case Diagram

Use case diagram merupakan suatu diagram yang menggambarkan interaksi antara sistem dengan *user* ataupun lainnya dalam memenuhi kebutuhan sistem. Simbol-simbol yang digunakan dalam *Use Case Diagram* menurut (Dennis et al., 2015), yaitu:

Tabel 2.4 Use Case Diagram

Sumber: (Dennis et al., 2015)

Simbol	Keterangan
	<p><i>An actor</i> atau aktor merupakan gambaran pihak-pihak yang berperan dalam sistem.</p>
	<p><i>Use case name</i> merupakan kegiatan yang dilakukan dalam sistem.</p>
	<p><i>System Boundary</i> merupakan sebuah kontak yang mewakili sebuah sistem.</p>
	<p><i>Association relationship</i> atau hubungan asosiasi memperlihatkan aktor mana saja yang terlibat dalam <i>use case</i>, dan bagaimana hubungan antara <i>use case</i> dengan <i>use case</i> lainnya. Hubungan antar <i>use case</i> digolongkan menjadi 2, yaitu: <i>extend</i> yang digambarkan dengan keterangan <<<i>extend</i>>>, dan <i>include</i> digambarkan dengan keterangan <<<i>include</i>>>.</p>

2.5.2. Class Diagram

Class diagram merupakan hubungan antar kelas yang tetap konstan dalam sistem dari waktu ke waktu. *Class diagram* menggambarkan kelas yang mencakup perilaku dan kedudukan dengan hubungan antar kelas. Simbol-simbol yang digunakan dalam *Class Diagram* menurut (Dennis et al., 2015), yaitu:

Tabel 2.5 Class Diagram

Sumber: (Dennis et al., 2015)

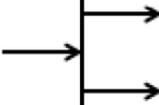
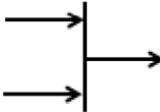
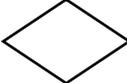
Simbol	Keterangan
<div style="border: 1px solid black; padding: 2px;"> <p style="text-align: center;">Class Name</p> <p>- <i>Attribute name</i></p> <p>- <i>/derived attribute name</i></p> <p>+ <i>Operation name ()</i></p> </div>	<p><i>A class</i> atau sebuah kelas yang menggambarkan jenis seperti orang, tempat atau hal lain yang harus disimpan oleh sistem.</p>
<p style="text-align: center;"><i>Attribute name</i> <i>/derived attribute name</i></p>	<p><i>A attribute</i> atau sebuah atribut yang merupakan gambaran keadaan suatu objek.</p>
<p style="text-align: center;"><i>Operation name ()</i></p>	<p><i>A method</i> atau sebuah metode yang dapat dilakukan oleh tindakan atau fungsi dari suatu kelas.</p>
<p style="text-align: center;"><u>1..* <i>verb phrase</i> 0..1</u></p>	<p><i>Association</i> atau asosiasi yang merupakan hubungan antara beberapa kelas, atau kelas itu sendiri.</p>

2.5.3. Activity Diagram

Activity diagram merupakan gambaran aktivitas yang terjadi dalam sebuah proses bisnis. *Activity diagram* dapat digunakan untuk menggambarkan alur kerja yang terdapat dalam *use case* hingga rincian spesifik dari *use case*. Simbol-simbol yang digunakan dalam *activity diagram* (Dennis et al., 2015), yaitu:

Tabel 2.6 Activity Diagram

Sumber: (Dennis et al., 2015)

Gambar	Keterangan
	<i>Activity</i> menggambarkan untuk mewakili serangkaian tindakan.
	<i>Action</i> menggambarkan untuk melakukan tindakan.
	<i>Initial node</i> merupakan objek dibentuk atau diawali.
	<i>Final node</i> merupakan objek dibentuk dan dihancurkan.
	<i>Fork node</i> merupakan salah satu aliran pada tahap tertentu yang berubah menjadi beberapa aliran.
	<i>Join node</i> merupakan salah satu aliran pada tahap tertentu yang menggabungkan beberapa aliran menjadi satu.
	<i>Decision</i> menggambarkan suatu keputusan yang harus diambil pada kondisi tertentu.

2.6. Aplikasi Mobile

Aplikasi *mobile* adalah aplikasi yang dapat digunakan walaupun pengguna berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadi pemutusan atau terputusnya komunikasi. Aplikasi ini dapat diakses melalui perangkat nirkabel seperti *pager*, seperti telepon seluler dan PDA. (Insani & Insanudin, 2016)

Aplikasi *mobile* yang merupakan sebuah aplikasi yang memungkinkan untuk melakukan mobilitas dengan menggunakan perlengkapan seperti PDA, telepon seluler atau *handphone*. Dengan menggunakan aplikasi *mobile*, maka dapat dengan mudah melakukan berbagai macam aktivitas mulai dari hiburan, berjualan, belajar, mengerjakan pekerjaan kantor, *browsing*, dan lain sebagainya.

Beberapa penelitian juga banyak yang menggunakan aplikasi *mobile*, baik untuk hiburan, mempermudah dalam layanan komunikasi data, maupun sebagai pengendali alat kamera DSLR. Aplikasi *mobile* dibangun dengan beberapa bahasa pemrograman *mobile*. Adapun contoh dari *mobile programming* untuk ponsel diantaranya adalah *Javafx mobile*, J2ME, C++, C#, .NET, dan *Flash Lite*. (Surahman & Setiawan, 2017)

2.7. Android

Android adalah sebuah sistem operasi untuk *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan platform yang terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Android merupakan generasi baru platform *mobile*, platform yang memberikan pengembang untuk melakukan pengembangan sesuai dengan yang diharapkannya. Berdasarkan pada pandangan tersebut, telah dibuat sebuah aplikasi kamus istilah komputer yang berbasis android sehingga masyarakat

bisa menggunakan kamus komputer secara *mobile*. (Mulyana & Maimunah, 2014)

Android yang merupakan sistem operasi bersifat *open source* berbasis linux dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya *Open*. Ponsel Android pertama mulai dijual pada tahun Oktober 2008. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance* yang merupakan konsorsium dari 34 perusahaan perangkat keras, perangkat lunak, dan telekomunikasi. (Surahman & Setiawan, 2017)

2.6. Decision Tree

Decision Tree atau yang biasa disebut pohon keputusan merupakan pendekatan “*divide and conquer*” dalam mempelajari masalah dari sekumpulan data independen yang digambarkan dalam bagan pohon. Metode klasifikasi yang melibatkan konstruksi pohon keputusan, koleksi node keputusan, terhubung oleh cabang-cabang, memperpanjang ke bawah dari simpul akar sampai berakhir di node daun. Dimulai dari node *root* yang oleh konvensi ditempatkan di bagian atas dari diagram pohon keputusan, atribut diuji pada node keputusan, dengan setiap hasil yang mungkin menghasilkan cabang. Setiap cabang kemudian mengarah ke node lain baik keputusan atau ke node

daun untuk mengakhiri. (Suryanto, Alfarobi, & Tutupoly, 2018; Wahyuningsih & Utari, 2018)

Decision tree merupakan struktur *flowchart* yang mempunyai *tree* (pohon), dimana setiap simpul internal menandakan suatu tes atribut, setiap cabang merepresentasikan hasil tes, dan simpul daun merepresentasikan kelas atau distribusi kelas. Alur pada *decision tree* ditelusuri dari simpul ke akar ke simpul daun yang memegang prediksi kelas. *Decision tree* adalah salah satu metode yang digunakan untuk pengklasifikasian dan prediksi karena memiliki kemudahan dalam interpretasi hasil. *Decision tree* dan algoritma C4.5 merupakan dua model yang tidak terpisahkan, oleh karena itu untuk membangun sebuah *decision tree*, dibutuhkan algoritma C4.5. Algoritma C4.5 merupakan pengembangan dari ID3. Beberapa pengembangan yang dilakukan C4.5 adalah bisa mengatasi *missing value*, *continue data*, dan *pruning*. Algoritma C4.5 mempunyai *input* berupa *training samples* dan *samples*, *training samples* berupa data contoh yang akan digunakan untuk membangun sebuah *tree* yang telah diuji kebenarannya, sedangkan *samples* merupakan *field-field* data yang nantinya akan kita gunakan sebagai parameter dalam melakukan klasifikasi data. (Rismayanti, 2018)

Decision tree memiliki kelebihan dan kekurangan. Berikut kelebihan dari *decision tree* menurut (Levine et al., 2015), yaitu:

- a. Memiliki akurasi yang baik.
- b. Dapat ditemukan kombinasi data yang tidak terduga.

- c. Integrasi yang mudah untuk masuk ke dalam sistem basis data.
- d. Dapat menghilangkan perhitungan yang tidak diperlukan karena sampel hanya diuji berdasarkan kriteria atau kelas tertentu.
- e. Area keputusan yang sebelumnya kompleks dan sangat global dapat dibuat menjadi lebih sederhana, dan lebih spesifik.
- f. Dapat pemilihan fitur yang fleksibel dari node internal yang berbeda, fitur yang dipilih membedakan kriteria dari kriteria lain di node yang sama.

Berikut kelemahan *decision tree* menurut (Levine et al., 2015), yaitu:

- a. Kesalahan akumulasi jumlah dari setiap level dalam pohon keputusan besar.
- b. Terjadi tumpang tindik ketika banyak kelas, dan kriteria yang digunakan.
- c. Kesulitan dalam merancang pohon keputusan yang optimal.
- d. Hasil kualitas dari keputusan yang diperoleh sangat tergantung bagaimana pohon itu dirancang.

2.7. Naive Bayes

Naive Bayes adalah salah satu metode yang populer dalam klasifikasi, metode ini juga terkenal dengan prediksi probabilitas dari data yang disajikannya. *Naive bayes* memiliki nama lain seperti *idiot's Bayes*, *simple Bayes*, dan *independence Bayes* karena kemudahannya serta tidak membutuhkan proses iterasi yang kompleks. *Naive Bayes* merupakan sebuah metode pengklasifikasian peluang sederhana dengan asumsi antar variabel penjelas saling bebas (independen) dengan menjumlahkan frekuensi dan

kombinasi nilai dari *dataset* yang diberikan. Keuntungan dari metode *naive bayes* adalah dapat hanya menggunakan data kecil untuk memperkirakan parameter *mean* dan *varians* dari variabel yang diperlukan untuk klasifikasi. Hal ini dikarenakan variabel independen yang diasumsikan hanya variabel untuk setiap kelas yang harus ditentukan, bukan seluruh matriks kovarians. (Amani, Maulana, & Syauqy, 2017; Iskandar & K. Suprpto, 2015; Oktanisa & Supianto, 2018)

Naive bayes memiliki kelebihan dan kekurangan. Berikut kelebihan dari *naive bayes* menurut (Oktanisa & Supianto, 2018), yaitu:

- a. Tidak diperlukan jumlah data yang banyak.
- b. Tidak diperlukan data *training* yang banyak.
- c. Bisa mengabaikan nilai yang hilang dalam perhitungan.
- d. Lama perhitungan termasuk cepat.
- e. Dapat digunakan untuk data kualitatif maupun kuantitatif.

Berikut kelemahan dari *naive bayes* menurut (Oktanisa & Supianto, 2018), yaitu:

- a. *Accuracy* yang tidak dapat diukur menggunakan satu probabilitas saja, sehingga menggunakan beberapa probabilitas untuk membuktikannya.
- b. Hasil prediksi probabilitas akan bernilai 0 (nol), jika ada kondisi probabilitas yang bernilai 0 (nol).

- c. Hanya bisa untuk mendeteksi kata-kata saja, tidak bisa berupa gambar.
- d. Asumsi masing-masing variabel independen dapat membuat akurasi berkurang. Hal ini dikarenakan adanya korelasi antar variabel.

2.8. Random Forest

Random Forest adalah metode *ensemble* dalam hal *learning* yang digunakan untuk klasifikasi, regresi, dan *task* lainnya. Metode *random forest* ini dapat berfungsi untuk proses pendugaan yang dilakukan dengan metode *bagging*. Kinerja dari *random forest* ini diadaptasi dari *decision tree*, dimana telah dilakukan *training* menggunakan sampel individu dan setiap atribut dipecah pada *tree* yang dipilih antaratribut *subset* yang bersifat acak. Pada proses klasifikasi di *random forest*, sampel individu didasarkan pada pilihan terbanyak dari kumpulan populasi *tree*, sehingga akhir model yang dihasilkan *random forest* didasarkan pada hasil dari keseluruhan *subset tree* yang telah dikembangkan. (Oktanisa & Supianto, 2018; Suryanto et al., 2018; Wibowo, 2015)

Random forest memiliki kelebihan dan kekurangan. Berikut kelebihan dari *random forest* menurut (Oktanisa & Supianto, 2018; Suryanto et al., 2018), yaitu:

- a. Dapat menangani ribuan variabel.
- b. Dapat memberikan perkiraan variabel apa yang penting dalam klasifikasi.

- c. Dapat memberikan metode yang efektif untuk memperkirakan data yang hilang.
- d. Dapat mempertahankan akurasi ketika sebagian besar data hilang.

Berikut kelemahan *random forest* menurut (Oktanisa & Supianto, 2018; Suryanto et al., 2018), yaitu:

- a. Klasifikasi yang dihasilkan sangat sulit untuk dipahami.
- b. Kompleksitas yang dimiliki *random forest* dapat memakan waktu dalam pengambilan keputusannya.
- c. Memerlukan jumlah data yang banyak.
- d. Lambatnya dalam membuat prediksi.
- e. Hasil prediksi yang dapat diragukan, bila banyaknya data yang hilang.

2.9. Penelitian Terdahulu

Penelitian terdahulu ini menjadi salah satu acuan dalam melakukan penelitian sehingga dapat memperkaya teori yang digunakan dalam mengkaji penelitian yang dilakukan. Dari penelitian terdahulu, tidak ditemukannya penelitian dengan judul yang sama seperti judul penelitian yang dilakukan, yaitu “Merekomendasi Buah Dalam Memenuhi Cairan Tubuh Harian Berbasis Aplikasi *Mobile*”. Dalam penelitian terdahulu, telah diangkat beberapa penelitian yang dapat dijadikan sebagai referensi dalam memperkaya bahan

kajian pada penelitian. Berikut merupakan penelitian terdahulu dari beberapa jurnal yang terkait dengan penelitian ini.

Tabel 2.8 Penelitian terdahulu

Nama Jurnal	PLoS ONE
Judul Jurnal	EXTERNAL VALIDATION AND COMPARISON OF THREE PEDIATRIC CLINICAL DEHYDRATION SCALES
Peneliti	Joshua Jauregui, Daniel Nelson, Esther Choo, Branden Stearns, Adam C. Levine, Otto Liebmann, Sachita P. Shah
Tahun	2014
Volume	9
Issue	5
Pembahasan	<p>Hasil Penelitian Dehidrasi tetap menjadi penyebab signifikan morbiditas dan mortalitas pada populasi anak di seluruh dunia. Di Amerika Serikat, gastroenteritis saja menyumbang lebih dari 1,5 juta kunjungan rawat jalan, 200.000 rawat inap dan 300 kematian setiap tahunnya. Penilaian yang akurat dari tingkat dehidrasi dapat membantu dokter memandu pengobatan dengan resusitasi cairan oral atau intravena, dan diperlukan untuk prognosis yang akurat dan manajemen sumber daya.</p> <p>Kesimpulan Skala Gorelick dan Skala Dehidrasi Klinis merupakan prediktor yang wajar dari dehidrasi pada anak-anak dengan diare atau muntah. Skala Organisasi Kesehatan Dunia dan gestalt dokter tidak membantu prediktor dehidrasi dalam kelompok.</p>

Nama Jurnal	Global Health Science and Practice
Judul Jurnal	EMPIRICALLY DERIVED DEHYDRATION SCORING AND DECISION TREEMODELS FOR

Judul Jurnal	CHILDREN WITH DIARRHEA: ASSESSMENT AND INTERNAL VALIDATION IN A PROSPECTIVE COHORT STUDY IN DHAKA, BANGLADESH
Peneliti	Adam C Levine, Justin Glavis-Bloom, Payal Modi, Sabiha Nasrin, Soham Rege, Chieh Chu, Christopher H Schmid, Nur H Alam
Tahun	2015
Volume	3
Issue	3
Pembahasan	<p>Hasil Penelitian Diare tetap menjadi salah satu kondisi paling umum dan paling mematikan yang mempengaruhi anak-anak di seluruh dunia. Menilai status dehidrasi secara akurat sangat penting untuk menentukan arah perawatan, namun tidak ada model diagnostik klinis untuk dehidrasi yang diturunkan secara empiris dan divalidasi untuk digunakan dalam rangkaian terbatas sumber daya.</p> <p>Kesimpulan Penelitian ini adalah yang penelitian pertama dalam memperoleh dan memvalidasi model diagnostik klinis yang akurat dan andal untuk dehidrasi dalam rangkaian terbatas sumber daya. Setelah <i>external validation</i>, <i>frontline providers</i> dapat menggunakan alat-alat baru ini untuk mengelola diare akut pada anak-anak dengan lebih baik.</p>

Nama Jurnal	International Journal of Sport Nutrition and Exercise Metabolism
Judul Jurnal	FLUID NEEDS FOR TRAINING, COMPETITION, AND RECOVERY IN TRACK AND FIELD ATHELES
Peneliti	Douglas J. Casa, Samuel N. Cheuvront, Stuart D. Galloway, and Susan M. Shirreffs
Tahun	2019

Volume	29
Issue	2
Pembahasan	<p>Hasil Penelitian Dehidrasi yang dapat merusak kinerja di sebagian besar acara Olimpiade, dan atlet harus terhidrasi dengan baik sebelum berolahraga. Jadi, <i>fluid</i> harus cukup dikonsumsi selama latihan untuk membatasi dehidrasi hingga kurang dari 2% dari massa tubuh dan juga harus memenuhi kebutuhan <i>sodium</i> ketika kehilangan keringat dalam jumlah banyak, terutama jika olahraga berlangsung lebih dari sekitar 2 jam. Selama pemulihan dari olahraga, rehidrasi harus mencakup penggantian air dan garam yang hilang karena keringat.</p> <p>Kesimpulan Dampak dehidrasi pada latihan dan hasil performa pada atlet tetap menjadi topik yang banyak diperdebatkan. Atlet lintasan dan lapangan sering melatih dan bersaing dalam kondisi lingkungan yang panas, di mana keseimbangan cairan dan hidrasi menjadi pertimbangan penting setiap hari. Melihat sifat individu dari respons berkeringat dengan pelatihan dan kompetisi, setiap atlet harus menilai kebutuhan cairan masing-masing dan menentukan apakah ini kemungkinan menjadi penyebab kekhawatiran kehilangan massa tubuh diamati. Risiko gangguan dalam pelatihan atau kinerja dengan tingkat dehidrasi <2% kehilangan massa tubuh adalah "rendah" dan berlaku untuk banyak peristiwa trek-dan-lapangan (terutama lari cepat, melompat, dan melempar). Namun, acara lintasan dan Lapangan lainnya memiliki risiko "tinggi", biasanya dalam durasi yang lebih lama dan kegiatan yang berkelanjutan, seperti acara ketahanan. Untuk acara-acara ini,</p>

Pembahasan	perhatian yang cermat harus diberikan pada praktik hidrasi individual dan terencana untuk mengoptimalkan hasil pelatihan dan kinerja.
-------------------	---

Nama Jurnal	ICIC Express Letters, Part B: Applications
Judul Jurnal	VACCINE PREDICTION SYSTEM USING ARIMA METHOD
Peneliti	Sahisnu, J.S., Natalia, F., Ferdinand, F.V., Sudirman, S., Ko, C.S.
Tahun	2020
Volume	11
Issue	6
Pembahasan	<p>Hasil Penelitian</p> <p>Telah dilakukan penelitian dengan strategi baru dalam memprediksi tingkat stok vaksin agar keperluan imunisasi dengan data dari Layanan Kesehatan Tangerang Selatan di Area Serpong 1 di Tangerang Selatan, Indonesia selama Januari 2018-Oktober 2018. Model yang digunakan adalah ARIMA Metode (1,1, 16) yang akan diimplementasikan di R Studio yang digunakan untuk membuat prediksi tingkat stok vaksin yang diperlukan untuk mengantisipasi kekurangan stok sebelum terjadi. Keakuratan model metode ARIMA (1, 1, 16) dihitung dengan menggunakan persentase kesalahan absolut rata-rata dan ditemukan 91,38%.</p> <p>Kesimpulan</p> <p>Adanya beberapa cara untuk memprediksi sistem, salah satunya dalam penelitian pengembangan sistem prediksi stok vaksin dengan membuat aplikasi berbasis R menggunakan metode ARIMA. Dalam penelitian ini, telah diterapkan di Kota Tangerang Selatan dengan menggunakan data dari Layanan Kesehatan Tangerang Selatan selama Januari 2018-Oktober 2018.</p>

Pembahasan	Dalam mengimplementasikan proses ilmu data dan memvisualisasikan hasil, digunakan perangkat lunak analitik data seperti Power BI, Tableau, dan R. Hasil yang didapatkan dari penelitian adalah tingkat keberhasilan 91,38% sehingga model ARIMA (1, 1, 16) dapat dikatakan lebih baik daripada model ARIMA (1, 1, 1) dan dapat digunakan untuk membuat prediksi
-------------------	---