

BAB II

LANDASAN TEORI

2.1 Penelitian Terkait

Pada tabel 2.1 merupakan penelitian yang berkaitan dengan skrining dengan menggunakan sebuah metode *Multi-Objective Optimization by Ratio Analysis (MOORA)* berbasis web.

Table 2. 1 Tabel Penelitian Terdahulu

1	Nama Jurnal	Jurnal Sistem Komputer dan Informatika (JSON)
	Judul Artikel	Sistem Pendukung Keputusan Perbandingan Metode MOORA Dengan MOOSRA Dalam Pemilihan <i>Hair Stylish</i>
	Penulis Artikel (Tahun)	Mohammad Aldinugroho Abdullah & Rima Tamara Aldisa (2023)
	Metode Pengembangan	Tahapan penelitian meliputi: Analisis masalah, pengumpulan data, studi literatur, penerapan metode (MOORA & MOOSRA), analisis perbandingan hasil, dan pembuatan laporan.
	Simpulan	Penelitian ini membandingkan metode MOORA dan MOOSRA untuk pemilihan <i>air stylish</i> . Hasilnya menunjukkan bahwa kedua metode sama-sama efektif dan memberikan peringkat tertinggi yang sama, yaitu pada alternatif 1 atas nama Poppy Sukma.
2	Nama Jurnal	CHAIN: Journal of Computer Technology, Computer Engineering and Informatics
	Judul Artikel	Sistem Pendukung Keputusan untuk Pemilihan Platform Pengembangan Aplikasi Web Menggunakan Metode MOORA
	Penulis Artikel (Tahun)	Reflan Nuari (2024)
	Metode Pengembangan	Tahapan penelitian terdiri dari: Identifikasi masalah dan penentuan kriteria, pengumpulan data alternatif, dan penerapan metode MOORA.
	Simpulan	Hasil perhitungan dengan metode MOORA menunjukkan bahwa React menempati peringkat pertama (nilai 0,4265), menjadikannya platform terbaik untuk pengembangan aplikasi web dalam

		kasus ini, diikuti oleh Angular (0,4261) dan Laravel (0,4142).
3	Nama Jurnal	Jurnal SISKOM-KB (Sistem Komputer dan Kecerdasan Buatan)
	Judul Artikel	Rancang Bangun Aplikasi Pengelolaan Stok Obat Berbasis Web dengan Pendekatan DSS Metode Moora (Studi Kasus Apotek XYZ)
	Penulis Artikel (Tahun)	Rudi Sutomo & Johny Hizkia Siringo Ringo (2022)
	Metode Pengembangan	Menggunakan kerangka berpikir yang meliputi <i>input</i> alternatif & kriteria, pembentukan matriks, normalisasi matriks, dan penentuan hasil preferensi.
	Simpulan	Penelitian ini berhasil membangun aplikasi pengelolaan stok obat untuk Apotek XYZ. Sistem ini memiliki fitur perangkingan dengan metode MOORA untuk membantu pengambilan keputusan pembelian stok obat, dengan hasil Parasetamol (ranking 1), Mylanta (ranking 2), dan Enstrostop (ranking 3).
4	Nama Jurnal	Jurnal JUREKSI (Jurnal Rekayasa Sistem)
	Judul Artikel	Implementasi Metode Moora Dalam Penerimaan Karyawan Baru (Studi Kasus : PT. Prima Multi Peralatan) Berbasis Web
	Penulis Artikel (Tahun)	Dedy Kurniawan Hutabarat & Charles Jhony Mantho Sianturi (2024)
	Metode Pengembangan	Menggunakan kerangka Fishbone yang terdiri dari tahapan: Analisis Kebutuhan, implementasi Metode MOORA, Penulisan Kode Program (Node.js & PostgreSQL), Pengujian Program, dan Pengguna.
	Simpulan	Sistem pendukung keputusan ini dapat membantu menentukan calon karyawan dan meminimalkan kesalahan dalam input data penilaian. Implementasi metode MOORA memberikan hasil keputusan yang lebih akurat, dengan Rafidah Hanim mendapatkan nilai tertinggi (0.252) dan diterima sebagai karyawan baru.

2.2 Tinjauan Teori

Landasan teori yang digunakan dalam penelitian ini disusun berdasarkan kajian dari sejumlah jurnal penelitian sebelumnya. Meskipun demikian, teori yang diadopsi tidak diambil secara menyeluruh, melainkan dipilih secara selektif sesuai dengan relevansi terhadap permasalahan yang diteliti. Hanya teori-teori yang memiliki keterkaitan langsung dengan fokus penelitian yang dimasukkan ke dalam laporan ini. Adapun teori yang dijadikan dasar meliputi konsep mengenai sistem pendukung keputusan, metode *Multi-Objective Optimization by Ratio Analysis (MOORA)*, Database serta Kegiatan donor darah yang menjadi objek kajian dalam penelitian ini. Dengan demikian, landasan teori yang disusun tidak hanya berfungsi sebagai acuan konseptual, tetapi juga sebagai pijakan dalam menganalisis data dan menarik kesimpulan penelitian.

2.2.1 *Database*

Database merupakan tempat penyimpanan kumpulan data dan informasi yang disusun secara metodis dan sistematis pada suatu tempat untuk memudahkan pengelolaan dan akses pada data yang ingin diubah, dianalisis dan diperbaharui dengan mudah, sehingga bisa diakses melalui program komputer untuk akses yang lebih mudah[13].

a. *PostGreSql*

PostgreSQL adalah sistem manajemen basis data relasional (*Relational Database Management System / RDBMS*) bersifat *open-source* dan berbasis objek relasional.[14]. Sedangkan PostgreSQL merupakan program pengelolaan serta penghubung antar aplikasi perangkat lunak dan *server database* yang mampu menerima dan mengirimkan suatu data dengan cepat[15].

Fungsi dari PostgreSQL pada dasarnya adalah untuk melakukan pengelolaan, menerima dan mengirimkan informasi atau data pada *database* yang sangat cepat dan *multi user* serta menggunakan perintah standar

PostgreSQL, Penerapan paling umum dari PostgreSQL adalah pembuatan *database* untuk sebuah *website*[16]. Adapun keistimewaan dari PostgreSQL sebagai berikut[17] .

1. PostgreSQL merupakan sistem manajemen basis data *open source* yang bisa digunakan oleh siapa saja tanpa biaya lisensi (Gratis). Hal ini menjadikannya pilihan utama bagi perusahaan kecil, pengembang, dan mahasiswa yang ingin membangun aplikasi dengan anggaran terbatas.
2. Kinerja Tinggi Dengan kinerjanya yang unggul, PostgreSQL dapat menjalankan sistem secara efisien bahkan pada spesifikasi hardware yang rendah. PostgreSQL tidak memerlukan banyak RAM, namun tetap dapat menangani kueri dengan cepat dan stabil.
3. PostgreSQL mendukung penggunaan oleh *multiple user* (multi-pengguna) tanpa mengorbankan kinerja. Selain itu, sistem ini skalabel sehingga dapat digunakan untuk segala hal mulai dari aplikasi kecil hingga sistem dengan basis data besar.
4. PostgreSQL memiliki sistem keamanan yang cukup baik, mulai dari otentikasi berbasis host, enkripsi kata sandi, hingga kontrol akses yang ketat. Hal ini memastikan data tetap aman dan terlindungi dari akses yang tidak sah.
5. PostgreSQL dilengkapi dengan berbagai fitur seperti tipe kolom, perintah dan fungsi, serta kemampuan tuning kinerja untuk mengoptimalkan kueri. Hal ini membuat PostgreSQL fleksibel untuk berbagai kebutuhan pemrosesan data.

b. Tipe Data PostgreSQL

Setiap pengembang dari PostgreSQL harus menentukan tipe data yang akan disimpan di setiap kolom saat membuat tabel. Berikut tipe dari data PostgreSQL serta contoh dari penggunaanya.

Tipe Data	Penjelasan	Contoh Penggunaan
INT	INT Digunakan untuk menyimpan data bilangan bulat (tanpa angka desimal) serta <i>user</i> bisa menentukan jumlah digit, Seperti INT(11) yang berarti bisa menampung 11 digit angka.	Kuantitas INT(11)
VARCHAR (size)	VARCHAR Digunakan untuk menyimpan data yang berisi text dengan panjang maksimal 225 karakter.	Nama VARCHAR(20)
TEXT	TEXT Digunakan untuk menyimpan text yang panjang yang mampu menyimpan karakter sebanyak antara 255 – 65535 huruf	Catatan TEXT
CHAR	CHAR Digunakan untuk menyimpan data yang panjangnya tidak lebih dari n huruf.	Status Ya/Tidak (Y/T) CHAR(1)

Table 2. 2 Tipe Data PostgreSQL

c. Neon

Neon merupakan sebuah produk layanan pihak ketiga atau bisa disebut sebagai *Platform-as-a-Service* (PaaS) dalam bentuk *Cloud Computing*. Layanan ini berbasis *platform* yang bersifat modular dengan ini pengguna tidak perlu melakukan proses pembangunan *database* dan pemeliharaan infrastruktur yang kompleks. Dengan kata lain, Neon menyediakan fondasi utama bagi pengembangan aplikasi tanpa harus melewati tahapan teknis yang rumit di sisi server.[18] Yang memungkinkan pengembangan aplikasi secara cepat dan berfokus lebih pada pengalaman pengguna aplikasi dan menyerahkan rangkaian proses sepenuhnya pada *cloud platform* untuk kebutuhan proses bisnis aplikasi kepada layanan pihak ketiga dalam bentuk PaaS[18]

Dengan demikian, Neon dapat diposisikan sebagai alternatif yang bagi pengembang yang membutuhkan layanan *backend* terintegrasi, mudah

digunakan, dan dapat diandalkan. Perannya sebagai *Platform-as-a-Service* membuktikan bahwa paradigma *Cloud Computing* tidak hanya mempermudah aspek teknis, tetapi juga mampu menghadirkan nilai tambah dalam konteks efisiensi, efektivitas, dan fokus pengembangan pada kebutuhan pengguna akhir.

2.2.2 Relational Database Management System (RDBMS)

Relational Database Management System (RDBMS) merupakan sistem manajemen basis data yang menggunakan model relasional untuk menyimpan, mengelola, dan memanipulasi data. Model relasional pertama kali diperkenalkan oleh E. F. Codd pada tahun 1970 melalui pendekatan matematis berbasis teori himpunan dan logika predikat, yang menjadikan data direpresentasikan dalam bentuk tabel (*relations*) yang saling terhubung melalui kunci (*keys*)[19]. Struktur tabel pada RDBMS terdiri dari baris (*tuples*) dan kolom (*attributes*), sehingga data dapat dikelola secara terstruktur dan konsisten.

RDBMS menyediakan mekanisme pengelolaan data yang menjamin integritas, konsistensi, dan keamanan, melalui penerapan berbagai aturan seperti *entity integrity*, *referential integrity*, serta *domain constraints*. Selain itu, RDBMS mendukung operasi manipulasi data menggunakan bahasa deklaratif Structured Query Language (SQL), yang memungkinkan pengguna melakukan operasi query, insert, update, dan delete dengan efisien[19], [20]. Kemampuan ini menjadikan RDBMS sebagai pilihan utama dalam pengembangan aplikasi yang membutuhkan transaksi data yang akurat, terkontrol, dan skalabel.

Beberapa karakteristik utama RDBMS meliputi dukungan terhadap normalisasi untuk mengurangi redundansi dan anomali data, kemampuan mengelola relasi antar tabel melalui primary key dan foreign key, serta

mendukung transaksi ACID (Atomicity, Consistency, Isolation, Durability) untuk memastikan keandalan data dalam lingkungan multi-user. RDBMS modern juga dilengkapi fitur indeks, View, stored procedure, dan mekanisme optimasi query untuk meningkatkan performa sistem.

Salah satu perbedaan RDBMS adalah DBMS menyimpan data dalam bentuk file, sedangkan RDBMS menyimpan data dalam bentuk tabel. Oleh karena itu, RDBMS menggunakan struktur tabel dengan headers sebagai nama kolom dan tiap barisnya terdapat values yang berkaitan dengan headers-nya. Selain dari perbedaan tipe data yang disimpan, RDBMS juga memiliki beberapa keunggulan yang membuat tipe database ini banyak digunakan

Dalam pengembangan sistem informasi berbasis web, RDBMS banyak digunakan karena kestabilan, fleksibilitas, serta kemampuannya dalam menangani data dalam jumlah besar secara terstruktur. Beberapa contoh RDBMS yang umum digunakan antara lain PostgreSQL, MySQL, MariaDB, SQL Server, dan Oracle[21]. Pada penelitian ini, PostgreSQL digunakan karena memiliki kepatuhan tinggi terhadap standar SQL, mendukung relasi kompleks, serta memiliki performa stabil untuk kebutuhan perhitungan dan pemrosesan data dalam sistem pendukung keputusan.

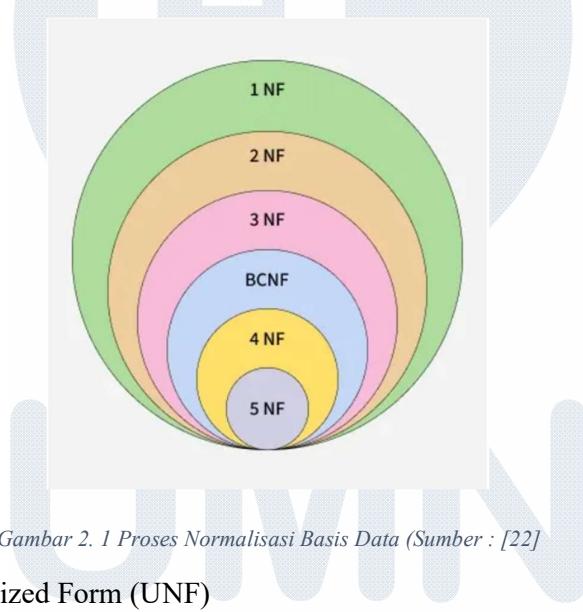
2.2.3.1. Normalisasi Basis Data

Normalisasi adalah teknik formal yang digunakan dalam pendekatan *bottom-up* untuk mengevaluasi dan memperbaiki struktur tabel basis data agar terhindar dari redundansi data serta anomali pada operasi *insert*, *update*, dan *delete*. Dalam pendekatan *bottom-up*, normalisasi berperan sebagai tahapan utama untuk mengelompokkan atribut data hasil pengumpulan data lapangan menjadi struktur tabel yang efisien dan konsisten[22]. Dengan melakukan normalisasi, basis data yang dihasilkan mampu merepresentasikan data secara akurat serta meminimalkan terjadinya anomali data pada proses penyimpanan dan pengolahan data.

Normalisasi basis data memiliki beberapa tujuan utama, antara lain:

1. Mengurangi redundansi atau duplikasi data dalam basis data.
2. Menghindari terjadinya anomali data, seperti anomali penyisipan, penghapusan, dan pembaruan data.
3. Meningkatkan konsistensi dan integritas data.
4. Mempermudah pemeliharaan dan pengembangan basis data.
5. Mengoptimalkan struktur tabel agar sesuai dengan kebutuhan sistem.

Dengan tercapainya tujuan-tujuan tersebut, basis data dapat mendukung pengolahan data dan pengambilan keputusan secara lebih efektif. Adapun tahapan normalisasi yang digunakan dalam penelitian ini meliputi [23]:



Gambar 2. 1 Proses Normalisasi Basis Data (Sumber : [22])

3. Unnormalized Form (UNF)

Unnormalized form adalah tahapan pertama untuk mengumpulkan atribut dan juga entitas yang memiliki satu atau lebih repeating groups. Tahap awal di mana seluruh atribut yang ditemukan dari formulir pemeriksaan pendonor (seperti Nama, Usia, Tekanan Darah, HB, dll.) dikumpulkan ke dalam satu kumpulan data besar tanpa memperhatikan format atau aturan tertentu.

4. First Normal Form (1NF)

First normal form (1NF) sebuah tabel dikatakan memenuhi 1NF jika setiap kolom hanya berisi nilai tunggal (atomik) dan tidak terdapat kelompok atribut yang berulang (*repeating groups*). Ada dua pendekatan yang biasanya dilakukan pada tahapan normalisasi pertama, yaitu mengisi bagian yang kosong dengan cara menduplikasi data yang tidak berulang, atau mengisi data yang berulang dengan copy dari atribut original key.

Pada tahap ini, dipastikan setiap data pendonor memiliki struktur yang seragam.

5. Second Normal Form (2NF)

Sebuah tabel memenuhi 2NF jika sudah berada dalam bentuk 1NF dan seluruh atribut non-kunci memiliki ketergantungan fungsional penuh terhadap kunci utama (*primary key*). Tahap ini bertujuan untuk menghilangkan ketergantungan parsial, di mana suatu atribut hanya bergantung pada sebagian dari kunci primer.

6. Third Normal Form (3NF)

Tabel dikatakan memenuhi 3NF jika sudah berada dalam bentuk 2NF dan tidak memiliki ketergantungan transitif (*transitive dependency*). Artinya, atribut non-kunci tidak boleh bergantung pada atribut non-kunci lainnya. Sebagai contoh, keterangan mengenai kriteria medis hanya boleh bergantung pada ID Kriteria, bukan pada ID Pemeriksaan Pendonor.

2.2.3 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan salah satu model konseptual yang digunakan dalam perancangan basis data untuk menggambarkan struktur data secara logis[24]. ERD digunakan untuk memodelkan hubungan antar data dengan merepresentasikan entitas, atribut, serta relasi yang terjadi di dalam suatu sistem. Model ini membantu perancang basis data dalam memahami kebutuhan data sebelum diimplementasikan ke dalam bentuk tabel relasional.

ERD berperan penting dalam tahap perancangan konseptual basis data karena mampu memberikan gambaran menyeluruh mengenai struktur data tanpa bergantung pada aspek teknis sistem manajemen basis data yang digunakan[25]. Dengan menggunakan ERD, proses komunikasi antara perancang sistem dan pengguna dapat dilakukan dengan lebih jelas dan terstruktur[26].

ERD memiliki peran penting dalam perancangan basis data karena menjadi dasar dalam proses normalisasi dan perancangan tabel relasional. Dengan ERD, perancang basis data dapat:

1. Mengidentifikasi kebutuhan data secara sistematis.
2. Menentukan struktur tabel dan relasi antar tabel.
3. Meminimalkan redundansi data.
4. Menjaga konsistensi dan integritas data.

Dalam penelitian ini, ERD digunakan sebagai acuan utama dalam merancang struktur basis data yang mendukung sistem pendukung keputusan penentuan kelayakan pendonor darah dan penerapan metode MOORA.

2.2.3.1. Entitas

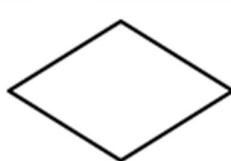
Entitas merupakan objek atau konsep dalam dunia nyata yang datanya perlu disimpan dalam basis data. Setiap entitas memiliki karakteristik tertentu yang membedakannya dari entitas lain. Dalam ERD, entitas biasanya direpresentasikan dalam bentuk persegi panjang. Pada sistem pendukung keputusan penentuan kelayakan pendonor darah, contoh entitas yang dapat diidentifikasi antara lain pendonor, pemeriksaan kesehatan, kriteria penilaian, dan hasil keputusan. Identifikasi entitas yang tepat sangat penting untuk memastikan bahwa seluruh kebutuhan data sistem dapat terakomodasi dengan baik[27], [28].



Gambar 2. 2 Simbol Entitas

2.2.3.2. Relasi

Relasi merupakan hubungan yang terjadi antara satu entitas dengan entitas lainnya. Relasi menunjukkan bagaimana data saling berhubungan dalam suatu sistem. Dalam ERD, relasi digambarkan dalam bentuk belah ketupat yang menghubungkan dua atau lebih entitas. Relasi dalam database sangat penting karena mencerminkan keterkaitan logis antar data. Pada sistem pendukung keputusan kelayakan pendonor darah, relasi dapat terjadi antara entitas pendonor dengan pemeriksaan kesehatan, serta antara pemeriksaan dengan hasil keputusan kelayakan[24].



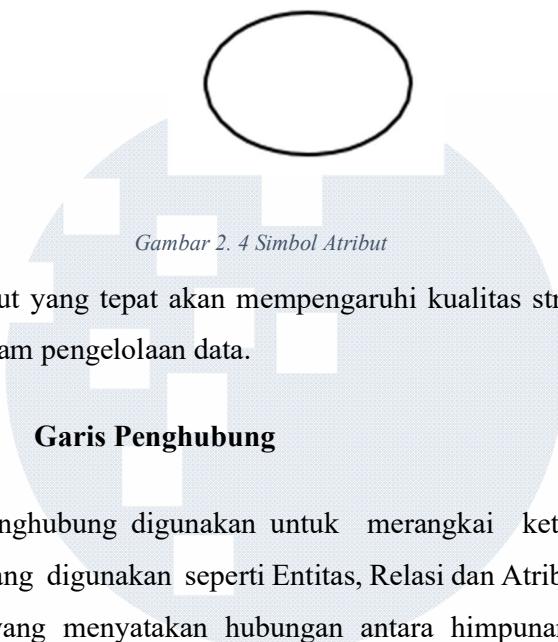
Gambar 2. 3 Simbol Relasi

2.2.3.3. Atribut

Atribut merupakan karakteristik atau properti yang dimiliki oleh suatu entitas. Atribut berfungsi untuk menjelaskan detail informasi dari entitas yang bersangkutan. Dalam ERD, atribut biasanya digambarkan dalam bentuk oval yang terhubung dengan entitasnya. Atribut dapat diklasifikasikan menjadi beberapa jenis, antara lain:

2. Atribut sederhana, yaitu atribut yang tidak dapat dipecah lagi.

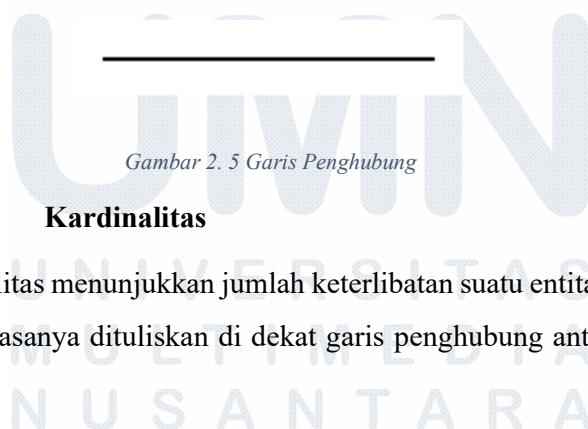
3. Atribut komposit, yaitu atribut yang dapat diuraikan menjadi beberapa subatribut.
4. Atribut kunci (key attribute), yaitu atribut yang digunakan untuk mengidentifikasi setiap entitas secara unik.



Pemilihan atribut yang tepat akan mempengaruhi kualitas struktur basis data dan kemudahan dalam pengelolaan data.

2.2.3.1. Garis Penghubung

Garis Penghubung digunakan untuk merangkai keterkaitan antara notasi-notasi yang digunakan seperti Entitas, Relasi dan Atribut. Disimbolkan dengan garis yang menyatakan hubungan antara himpunan relasi dengan himpunan entitas dan atribut dengan himpunan entitas.



2.2.3.2. Kardinalitas

Kardinalitas menunjukkan jumlah keterlibatan suatu entitas dalam relasi. Kardinalitas biasanya dituliskan di dekat garis penghubung antara entitas dan relasi.

Dalam hubungan antar entitas, dibagi menjadi berbagai macam diantaranya :

- a. One-to-One (1:1), yaitu satu entitas hanya berhubungan dengan satu entitas lainnya.

Relasi One To One



atau



Gambar 2. 6 Relasi One-to-One (Sumber: [28])

- b. One-to-Many (1:N), yaitu satu entitas dapat berhubungan dengan banyak entitas lainnya.

Relasi One To Many



Gambar 2. 7 Relasi One-to-Many (Sumber: [28])

- c. Many-to-Many (M:N), yaitu banyak entitas dapat saling berhubungan.



Gambar 2. 8 Relasi Many-to-Many (Sumber: [28])

Penentuan kardinalitas yang tepat sangat penting karena akan mempengaruhi struktur tabel pada tahap perancangan logis basis data.

2.2.4 Web

Dalam pengertiannya *website* merupakan kumpulan halaman yang saling terhubung secara logis dan dapat diakses melalui satu URL (*Uniform Resource Locator*), di dalamnya berisi informasi data digital baik berupa teks, gambar, animasi, suara dan video atau gabungan dari semuanya yang disediakan melalui jalur koneksi internet sehingga dapat diakses dan dilihat oleh semua orang di seluruh dunia[29].

Halaman *website* pada penelitian ini dibangun menggunakan bahasa standar yaitu Node.js. Kode Node.js tersebut kemudian diterjemahkan oleh web browser sehingga dapat ditampilkan dalam bentuk informasi yang dapat diakses dan dibaca oleh semua orang. Secara umum, *website* memiliki beberapa fungsi, antara lain[29]:

1. Memudahkan pengguna dalam melakukan komunikasi
2. Menjadi sumber mata pencaharian karena dapat dimanfaatkan sebagai sarana bisnis jual beli barang maupun jasa.

2.2.5 Bahasa Pemrograman (Java Script)

JavaScript merupakan bahasa pemrograman utama yang digunakan dalam penelitian pengembangan *Website* Sistem Pendukung Keputusan (SPK) pendonor darah ini, baik pada sisi *frontend* maupun *backend*. JavaScript awalnya merupakan bahasa yang berjalan di browser, namun melalui perkembangan teknologi muncul beberapa *platform* seperti Node.js, Express.js dan Next.js. JavaScript kini dapat digunakan untuk membangun aplikasi server side secara penuh[30].

Dalam penelitian ini, JavaScript digunakan untuk mengimplementasikan logika perhitungan metode MOORA, melakukan pemrosesan data, membuat API *backend*, serta membangun tampilan *frontend*. Dengan karakteristik yang bersifat *asynchronous* dan *event driven*, JavaScript mampu menangani proses secara efisien tanpa menghambat eksekusi program lainnya. Selain itu, ekosistem JavaScript yang luas, termasuk dukungan library dan modul NPM, mempermudah integrasi antara *frontend*, *backend*, dan database PostgreSQL. Keseluruhan kemampuan tersebut menjadikan JavaScript bahasa yang fleksibel dan efektif untuk membangun aplikasi SPK berbasis web yang responsif, dinamis, dan mudah dikembangkan.

Pada penelitian ini digunakan beberapa *platform* dari JavaScript yang dijelaskan sebagai berikut.

a. Node.js

Node.js berfungsi sebagai *runtime environment* yang mengeksekusi seluruh kode JavaScript pada sisi *backend* dan server API. Node.js menyediakan performa tinggi melalui arsitektur *non-blocking I/O* sehingga mampu menangani permintaan API secara cepat, termasuk perhitungan metode MOORA untuk menentukan kelayakan pendonor darah[31]. Node.js berperan sebagai fondasi utama yang menangani seluruh proses *backend*, mulai dari pengolahan data, pengelolaan request-response, hingga integrasi dengan database PostgreSQL. Arsitektur Node.js yang menggunakan *single threaded event loop* membuat aplikasi mampu menangani banyak permintaan secara efisien tanpa harus membuat banyak thread, sehingga performanya lebih ringan dan responsif. Selain itu, ekosistem NPM yang luas memudahkan pengembang untuk menggunakan berbagai library, termasuk Express.js sebagai framework *backend*. Melalui Node.js, sistem SPK dapat menjalankan proses komputasi seperti normalisasi

data dan perhitungan metode MOORA secara cepat, sekaligus menyediakan API yang stabil untuk diakses oleh *frontend* Next.js. Dengan demikian, Node.js menjadi komponen inti yang memastikan aplikasi ini bekerja secara optimal dan mampu menangani proses pengolahan data secara real-time.

b. Node.js

Express.js merupakan sebuah framework backend berbasis Node.js yang digunakan untuk membangun layanan server dan API secara efisien[32]. Pada *Website* Sistem Pendukung Keputusan (SPK) pendonor darah ini, Express.js menyediakan struktur yang sederhana namun fleksibel untuk menangani proses *routing*, manajemen *request response*, serta penggunaan *middleware* yang mendukung berbagai kebutuhan seperti validasi input, autentikasi, hingga pengolahan data sebelum dikirimkan ke klien. Dengan memanfaatkan Express.js, pengembangan backend menjadi lebih terorganisir karena setiap fungsi server dapat diatur melalui endpoint yang jelas dan modular. Framework ini juga mempermudah integrasi dengan database PostgreSQL serta eksekusi logika perhitungan metode MOORA di sisi server. Keunggulannya yang ringan, cepat, dan mudah dikembangkan membuat Express.js sangat cocok sebagai fondasi layanan backend yang stabil dan responsif, sehingga mampu memenuhi kebutuhan aplikasi SPK berbasis web yang membutuhkan pemrosesan data secara real-time[32].

c. Next.js

Next.js adalah framework *frontend* berbasis React.js yang digunakan untuk membangun antarmuka dari *Website* Sistem Pendukung Keputusan (SPK) pendonor darah. Next.js menyediakan lingkungan pengembangan yang modern dengan dukungan *server-side*

rendering (SSR), *static site generation* (SSG), dan pengelolaan routing yang otomatis, sehingga tampilan web dapat dimuat lebih cepat, responsif, dan optimal di berbagai perangkat[33]. Pada penelitian ini, Next.js digunakan untuk mengimplementasikan halaman-halaman utama, menampilkan hasil perhitungan metode MOORA, serta menyediakan pengalaman pengguna yang interaktif melalui komponen-komponen React. Selain itu, Next.js terintegrasi secara langsung dengan API backend yang dibangun menggunakan Express.js, sehingga proses pengambilan data, validasi, dan penyajian informasi pendonor dapat dilakukan dengan efisien. Kemampuan Next.js dalam menggabungkan rendering sisi klien dan server menjadikannya pilihan ideal untuk membangun sistem SPK berbasis web yang membutuhkan performa tinggi, struktur kode yang terorganisir, serta pengalaman pengguna yang intuitif[33].

2.2.6 Sistem Pendukung Keputusan

Sistem Pendukung Keputusan atau *Management Decision System* dikemukakan oleh Michael S.Scot Morton dan G.Antony Gorry pada tahun 1960-an. Pada tahun 1971, istilah *Decision Support System* (DSS) atau Sistem Pendukung Keputusan (SPK) mulai diperkenalkan. Sistem ini didefinisikan sebagai sistem berbasis komputer yang dirancang untuk memanfaatkan data serta model tertentu dengan tujuan memberikan dukungan kepada pengambil keputusan. Melalui pemanfaatan sistem tersebut, proses identifikasi dan penyelesaian masalah dapat dilakukan lebih terarah, sehingga keputusan yang diambil menjadi lebih tepat dan sesuai dengan kebutuhan [34].

Tujuan utama dari proses pengolahan data dan informasi dalam sistem pendukung keputusan adalah menghasilkan berbagai alternatif keputusan yang dapat dipertimbangkan oleh manajemen. Dengan demikian, sistem pendukung keputusan diposisikan sebagai salah satu bentuk penerapan sistem informasi

yang berperan sebagai alat bantu, bukan sebagai pengganti, dalam proses pengambilan keputusan. Artinya, sistem ini tidak menggantikan peran utama pengambil keputusan, melainkan memberikan dukungan berupa informasi, analisis, dan rekomendasi yang relevan untuk memperkuat kualitas keputusan yang dihasilkan.

Selain itu, sistem pendukung keputusan dirancang agar mampu menyajikan berbagai alternatif solusi yang dapat digunakan oleh pengambil keputusan dalam menyelesaikan masalah yang dihadapi. Peran utamanya adalah meningkatkan efektivitas sekaligus efisiensi dalam kegiatan manajerial, terutama dalam konteks proses pengambilan keputusan yang kompleks. Lebih jauh, sistem ini juga memadukan kemampuan komputer dalam memberikan pelayanan interaktif kepada penggunanya dengan proses pengolahan atau manipulasi data menggunakan model maupun aturan yang bersifat tidak terstruktur. Hasil akhirnya berupa alternatif keputusan yang sesuai dengan situasi tertentu, sehingga dapat memberikan nilai tambah yang signifikan dalam mendukung keberhasilan organisasi.

2.2.7 Karakteristik dan Kemampuan Sistem Pendukung Keputusan

Terdapat sejumlah karakteristik yang dimiliki oleh sistem pendukung keputusan, yang di antaranya dapat dijelaskan sebagai berikut [35].

1. Memberikan dukungan terhadap keseluruhan aktivitas organisasi.
2. Mampu membantu pengambilan beberapa keputusan yang saling berhubungan.
3. Bersifat konstan dan dapat digunakan secara berulang.
4. Terdiri atas dua komponen utama, yaitu data dan model.
5. Memanfaatkan data baik dari sumber internal maupun eksternal.
6. Memiliki kemampuan untuk melakukan *what-if analysis* serta *goal seeking analysis*.

7. Menggunakan beragam model kuantitatif dalam proses pengolahan informasi.

Proses pengambilan keputusan tidak dapat dilakukan secara sembarangan, melainkan memerlukan langkah-langkah yang sistematis agar keputusan yang dihasilkan benar-benar tepat dan efektif. Tahapan pengambilan keputusan terdiri dari beberapa langkah sebagai berikut[35].

1. Melakukan identifikasi terhadap masalah yang muncul.
2. Menentukan metode yang tepat untuk menyelesaikan masalah tersebut.
3. Mengumpulkan data yang diperlukan guna menjalankan model keputusan.
4. Menerapkan model keputusan yang telah dipilih.
5. Mengevaluasi kelebihan serta potensi dari setiap alternatif yang tersedia.
6. Melaksanakan solusi yang dipilih sebagai keputusan akhir.

Dalam perkembangannya, sistem pendukung keputusan tidak hanya dipahami dari konsep dasarnya, tetapi juga dari kemampuan serta keterbatasan yang dimilikinya. Kemampuan ini menunjukkan sejauh mana sistem dapat mendukung manajer atau pengambil keputusan dalam menyelesaikan masalah yang kompleks. Namun demikian, sistem pendukung keputusan juga memiliki batasan tertentu sehingga tidak bisa sepenuhnya menggantikan peran manusia. Karena itu, penting untuk memahami keunggulan sekaligus keterbatasannya agar pemanfaatan sistem ini dapat dilakukan secara optimal sesuai kebutuhan organisasi.

Berikut poin-poin penjelasan kemampuan yang seharusnya dimiliki oleh sistem pendukung keputusan antara lain[36].

1. Mendukung pengambilan keputusan manajerial dalam menangani masalah semi-terstruktur maupun tidak terstruktur.

2. Membantu manajer pada seluruh tingkatan manajemen, mulai dari level atas hingga level bawah.
3. Menunjang pengambilan keputusan baik secara individu maupun kelompok.
4. Memfasilitasi pengambilan keputusan yang saling bergantungan dan berurutan.
5. Mendukung seluruh tahapan proses pengambilan keputusan, mulai dari *intelligence, design, choice, hingga implementation*.
6. Mampu menangani berbagai bentuk proses serta jenis keputusan yang berbeda.
7. Bersifat fleksibel dan mampu beradaptasi dengan cepat terhadap perubahan.
8. Memberikan kemudahan interaksi antara sistem dan pengguna.
9. Lebih menekankan pada peningkatan efektivitas dibandingkan efisiensi dalam pengambilan keputusan.
10. Mudah dikembangkan oleh pengguna akhir sesuai kebutuhan.
11. Memiliki kemampuan pemodelan serta analisis dalam proses pengambilan keputusan.
12. Mempermudah akses terhadap berbagai sumber data dengan format yang beragam.

Di samping itu, sistem pendukung keputusan juga memiliki sejumlah keterbatasan di antaranya[36].

1. Tidak semua kemampuan manajerial maupun bakat manusia dapat dimodelkan, sehingga model dalam sistem tidak sepenuhnya mencerminkan kondisinya.
2. Kapasitas sistem terbatas pada pengetahuan dasar serta model yang tersedia.
3. Proses yang dijalankan sangat bergantung pada kemampuan *Software* yang digunakan.
4. Sistem pendukung keputusan tidak memiliki intuisi sebagaimana manusia, karena pada dasarnya hanya berupa kombinasi perangkat keras, perangkat lunak, dan sistem operasi tanpa kemampuan berpikir mandiri.

2.2.8 Multi-Criteria Decision Making (MCDM)

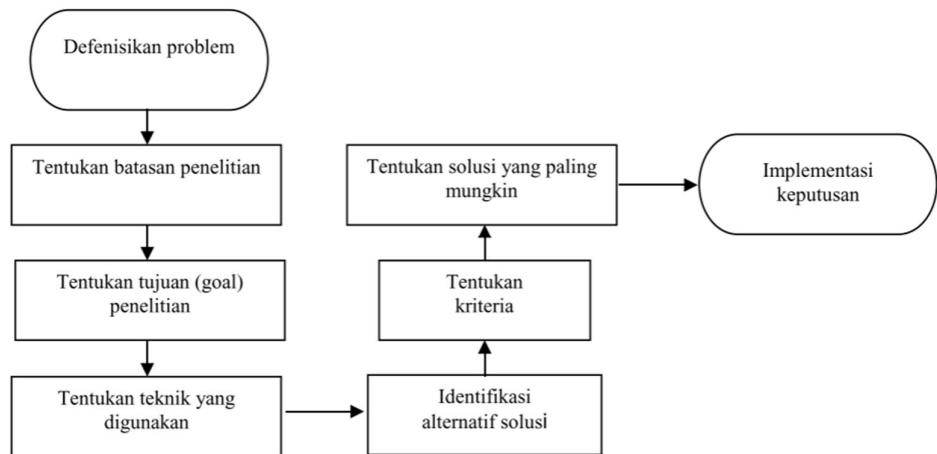
Multi-Criteria Decision Making (MCDM) Merupakan pendekatan atau metode dalam pengambilan keputusan berdasarkan opsi atau alternatif dari multi kriteria, Secara fundamental, MCDM termasuk dalam disiplin ilmu riset operasi. Pembahasannya meliputi pertimbangan baik aspek kualitatif maupun kuantitatif, serta penerapan berbagai kriteria yang terkadang saling bertentangan dalam mencapai target kinerja, sehingga memerlukan proses normalisasi terhadap nilai dari setiap kriteria yang ada[37].

Permasalahan yang melibatkan evaluasi beberapa kriteria dan alternatif, biasanya, dianggap sebagai bagian dari domain *Multi-Criteria Decision Making* (MCDM). Permasalahan di area ini secara umum dapat diklasifikasikan menjadi dua kategori:

1. *Multiple attribute decision making* (MADM) hal ini yang mempertimbangkan sejumlah alternatif yang telah ditentukan sebelumnya dan peringkat preferensi diskrit
2. *Multiple objective decision making* (MODM) yang mengakomodasi permasalahan terkait desain dan perencanaan untuk mendapatkan solusi optimal dari serangkaian tujuan untuk serangkaian interaksi yang berlaku di antara kendala.

Proses analisis *Multi-Criteria Decision Making* (MCDM) diatur oleh tiga langkah utama yaitu [37], [38]:

1. Identifikasi kriteria dan alternatif yang relevan dari teori dan praktik yang ada.
2. Pemberian nilai numerik pada kriteria untuk menunjukkan kepentingan relatifnya dan untuk mengkuantifikasi dampak alternatif terhadap kriteria tersebut.
3. Penggunaan prosedur matematika formal untuk menganalisis nilai numerik guna menentukan peringkat (prioritas) alternatif.



Gambar 2. 9 Proses analisis Multi-Criteria Decision Making (MCDM)

2.2.9 Metode *Multi-Objective Optimization by Ratio Analysis* (MOORA)

Metode *Multi-Objective Optimization by Ratio Analysis* (MOORA) adalah salah satu teknik dalam *Multiple Criteria Decision Making* (MCDM) yang diaplikasikan untuk mengoptimalkan dua atau lebih kriteria atau atribut yang sering kali memiliki konflik secara bersamaan. Metode ini pertama kali diperkenalkan oleh Brauers dan Zavadskas pada tahun 2006 sebagai bagian dari upaya untuk mengatasi keterbatasan metode-metode pengambilan keputusan sebelumnya, terutama dalam konteks isu industri dan manajerial yang rumit[39].

Konsep dasar metode MOORA adalah melakukan normalisasi terhadap nilai setiap kriteria agar seluruh data berada pada skala yang sama, kemudian mengoptimalkan nilai kriteria bertipe manfaat (benefit) dan meminimalkan nilai kriteria bertipe biaya (cost). Proses ini menghasilkan nilai optimasi akhir yang disebut sebagai nilai Y_i . Nilai Y_i yang semakin besar menunjukkan alternatif yang semakin baik. Normalisasi dalam MOORA dilakukan untuk menghilangkan pengaruh perbedaan satuan antar kriteria, sehingga setiap kriteria memiliki kontribusi yang seimbang dalam proses pengambilan keputusan. Dengan demikian, keputusan yang dihasilkan menjadi lebih adil dan

objektif. Pada dasarnya, metode MOORA beroperasi dengan cara menyesuaikan nilai untuk setiap kriteria yang ada. Selanjutnya, metode ini melakukan perhitungan rasio perbandingan antara karakteristik yang memberikan keuntungan (*benefit*) dengan karakteristik yang bersifat memberikan beban (*cost*). Keluaran dari tahapan ini adalah sebuah nilai gabungan (skor komposit) yang dapat dimanfaatkan untuk mengidentifikasi alternatif terunggul dari sekumpulan opsi yang tersedia. Dengan demikian, MOORA dapat menyediakan suatu penyelesaian yang lebih berdasar pada fakta karena memperhatikan seluruh kriteria secara bersamaan[39].

Keunggulan utama metode MOORA adalah kesederhanaan perhitungan, kejelasan konsep, serta fleksibilitas penerapannya. Metode ini dapat diaplikasikan pada berbagai bidang, seperti[40]:

1. Manajemen dan Ekonomi – untuk mendukung pengambilan keputusan strategis pada perusahaan.
2. Konstruksi dan Teknik – dalam penentuan prioritas proyek, pemilihan material, hingga evaluasi risiko.
3. Sistem Informasi dan Teknologi – untuk mendukung sistem pendukung keputusan (Decision Support System/DSS).
4. Bidang Sosial dan Kesehatan– misalnya dalam penentuan kelayakan pendonor darah

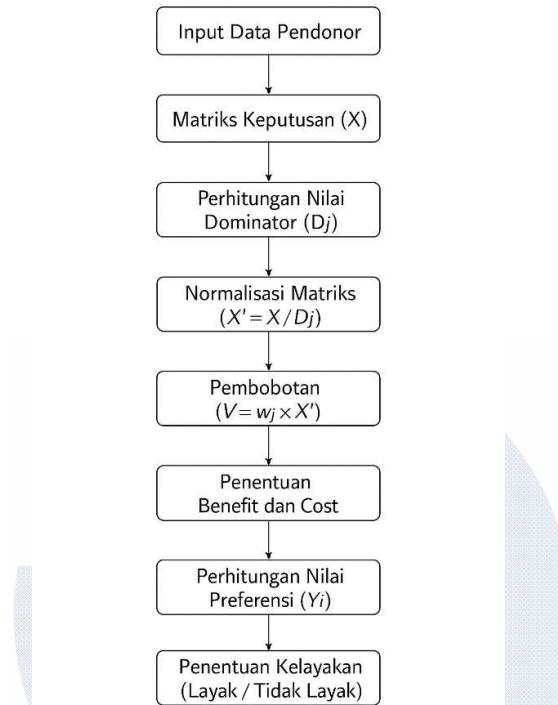
MOORA dinilai efektif dalam menyelesaikan masalah pengambilan, maka berikut ini merupakan keunggulan dari Metode MOORA dalam pengambilan keputusan[38]. Keputusan karena perhitungannya yang relatif sederhana namun tetap mampu mengakomodasi kompleksitas kriteria yang beragam hal ini disebabkan MOORA dalam pengambilan keputusan dapat dengan mudah menerapkan metode ini untuk mengevaluasi alternatif dan memilih yang paling sesuai, tanpa perlu memahami makna fisik dari proses pengambilan keputusan.

Metode ini banyak digunakan dalam penelitian maupun praktik, khususnya ketika pengambil keputusan dihadapkan pada *trade-off* antar kriteria yang tidak bisa diabaikan. Berbeda dengan metode *MCDM (Multi-Criteria Decision Making)* lainnya, MOORA menggunakan model matematis terpisah untuk kriteria manfaat dan non-manfaat, serta mengakomodasi kriteria kualitatif grafis dari matriks keputusan. Karena keunggulan tambahan ini, dalam metode MOORA, kemungkinan kehilangan informasi sangat kecil.

Tetapi Metode MOORA juga terdapat berbagai batasan yaitu[38].

Khususnya saat menetapkan bobot untuk alternatif, cukup sulit untuk memprediksi bobot yang diberikan pada berbagai kriteria. Dalam situasi tertentu, jumlah perbandingan pasangan atribut dan alternatif yang tidak terkelola terhadap setiap atribut membuat proses menjadi sangat kompleks. Serta Seiring bertambahnya jumlah alternatif, jumlah perhitungan meningkat dengan cepat dan prosedur komputasi menjadi sangat rumit. Secara garis besar, tahapan penyelesaian masalah dengan metode MOORA meliputi[38], [41]:





Gambar 2. 10 Proses Perhitungan Metode MOORA

1. Identifikasi kriteria.

Langkah pertama penyelesaian metode MOORA merupakan identifikasi kriteria, maksud dari langkah dibutuhkan untuk menentukan tujuan, mengidentifikasi atribut evaluasi yang bersangkutan dan *input* bobot nilai kriteria.

2. Penyusunan matriks keputusan.

Tahap berikutnya penerapan MOORA dilakukan dengan menyusun matriks keputusan yang berfungsi sebagai pengukur kinerja dari alternatif j pada atribut i . Dalam hal ini, setiap baris pada matriks merepresentasikan satu alternatif, sedangkan setiap kolom mewakili satu kriteria seorang calon pendonor darah. Nilai yang digunakan pada matriks ini diperoleh dari hasil konversi subkriteria, misalnya tekanan darah yang diklasifikasikan ke dalam skala 1–5, kadar

hemoglobin, atau lama waktu tidur. Proses ini bertujuan untuk mengubah data kualitatif menjadi bentuk kuantitatif agar dapat diolah secara matematis. Secara matematis, matriks keputusan dinyatakan sebagai berikut.

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{1n} \\ x_{21} & x_{22} & x_{2n} \\ x_{n1} & x_{n2} & x_{mn} \end{bmatrix}$$

Keterangan :

X_{ij} = respon alternatif ke- j pada kriteria ke- i

$i=1,2,3,\dots, n$ adalah nomor urutan atribut atau kriteria

$j=1,2,3,\dots, m$ adalah nomor urutan alternatif

X = Matriks keputusan

3. Perhitungan Nilai Dominator.

Tahapan selanjutnya pada perhitungan metode MOORA adalah menghitung nilai dominator. Nilai dominator merupakan nilai bilangan penyebut dalam proses normalisasi matrix pada tahapan berikutnya agar setiap kriteria memiliki skala yang seragam yang artinya nilai dominator di fungsikan sebagai penyeimbang perbedaan skala antar kriteria sehingga tidak ada satu kriteria yang mendominasi hasil akhir. Dalam metode MOORA, nilai dominator yang diperoleh dari dataset awal ditetapkan sebagai nilai tetap (fixed). Artinya, ketika data baru masuk ke sistem, proses normalisasi akan menggunakan nilai dominator yang sama tanpa menghitung ulang seluruh dataset.

Pendekatan ini memberikan konsistensi dan efisiensi, terutama untuk sistem pengambilan keputusan yang berjalan secara dinamis atau berkelanjutan (real-time).

time). Perhitungan dilakukan dengan cara menghitung akar dari jumlah kuadrat seluruh nilai pada kolom ke- j , sesuai rumus.

$$D_j = \sqrt{\sum_{i=1}^m x_{ij}^2}$$

Keterangan :

D_j = nilai dominator pada kriteria ke- j

x_{ij} = nilai dari alternatif

m = Jumlah total alternatif

4. Normalisasi matriks keputusan.

Langkah berikutnya dalam penerapan metode MOORA yaitu melakukan normalisasi matrix keputusan. Tujuan dari normalisasi adalah untuk menyamakan skala setiap elemen dalam matriks sehingga semua nilai menjadi seragam. Perhitungan tahap ini dilakukan menggunakan rumus berikut.

$$x'_{ij} = \frac{x_{ij}}{\sqrt{D_j}}$$

Keterangan :

x'_{ij} = nilai ternormalisasi dari alternatif ke- i pada kriteria ke- j

x_{ij} = nilai asli dari alternatif

D_j = nilai dominator pada kriteria ke- j

5. Pembobotan (*Weighted Normalized Matrix*).

Setelah perhitungan normalisasi matrix keputusan, selanjutnya dalam perhitungan metode MOORA adalah dengan melakukan Pembobotan dilakukan untuk menyesuaikan pengaruh relatif setiap kriteria terhadap keputusan akhir. Pembobotan dilakukan dengan mengalikan nilai normalisasi x'_{ij} dengan bobot kriteria w_j , sebagaimana ditunjukkan pada Persamaan.

$$V_{ij} = w_j \times x'_{ij}$$

Keterangan :

V_{ij} = nilai hasil perkalian antara bobot dan nilai normalisasi

w_j = bobot dari kriteria

x'_{ij} = nilai ternormalisasi dari alternatif ke- i pada kriteria ke- j

Hasil pembobotan ini menghasilkan matriks tertimbang (V) yang menjadi dasar untuk menghitung nilai preferensi. Kriteria dengan bobot besar (misalnya kadar hemoglobin dan tekanan darah) akan memiliki pengaruh yang lebih kuat terhadap hasil akhir.

6. Perhitungan Kriteria Benefit dan Cost.

Pada metode MOORA, setiap kriteria dibedakan menjadi dua jenis kriteria, yaitu Benefit dan Cost. Kriteria yang dimaksud dengan benefit merupakan kriteria yang nilainya semakin tinggi maka semakin baik, sedangkan kriteria cost adalah kriteria yang nilainya semakin kecil maka semakin baik.

Penentuan tipe ini penting agar sistem dapat melakukan penghitungan nilai preferensi secara benar, di mana nilai cost nantinya akan dikurangkan dari total nilai benefit.

Pada penelitian ini:

- Benefit: C1 (Tekanan darah), C2 (Berat badan), C3 (Hemoglobin), C4 (Konsumsi obat terakhir), C6 (Lama tidur), dan C7 (Riwayat penyakit).
- Cost: C5 (Umur), karena semakin muda usia pendonor, semakin optimal kondisi tubuh untuk mendonorkan darah.

7. Perhitungan nilai optimasi (*optimizing score*).

Tahap terakhir dalam metode MOORA adalah menghitung nilai optimasi (Y_i score) nilai ini merupakan nilai akhir untuk menentukan kelayakan calon pendonor darah. Nilai optimasi ini diperoleh dari hasil selisih antara jumlah nilai benefit dan nilai cost tertimbang. Perhitungan tahap ini dilakukan menggunakan rumus berikut:

$$Y_i = \sum_{j \in Benefit} V_{ij} - \sum_{j \in Cost} V_{ij}$$

Keterangan :

Y_i = nilai optimasi metode MOORA

V_{ij} = nilai hasil perkalian antara bobot dan nilai normalisasi

Benefit = himpunan kriteria dengan sifat menguntungkan (semakin besar semakin baik)

Cost = himpunan kriteria dengan sifat merugikan (semakin kecil semakin baik)

8. Penentuan Kelayakan dan Peringkat.

Susunan peringkat berdasarkan nilai Y_i menunjukkan hasil keputusan akhir. Alternatif dengan nilai Y_i tertinggi ditetapkan sebagai pilihan terbaik, sedangkan alternatif dengan nilai Y_i terendah ditetapkan sebagai pilihan yang kurang optimal.

Dalam penelitian ini, penentuan kelayakan dilakukan dengan membandingkan nilai Y_i terhadap ambang batas (*threshold*) yang ditetapkan. Ambang batas (*threshold*) pada penelitian ini menggunakan persentil ke-25 (P25) dari distribusi seluruh nilai Y_i . Artinya calon pendonor dengan nilai $Y_i \geq$ P25 dikategorikan sebagai pendonor yang layak sedangkan calon pendonor dengan nilai $Y_i < P25$ dikategorikan sebagai pendonor yang tidak layak.

Penggunaan persentil ke-25 adalah hasil dari observasi lapangan dan wawancara dengan pihak pelaksana pendonoran darah di lapangan, ditemukan bahwa pada hasil perangkingan dengan metode MOORA terdapat sebagian pendonor yang memiliki nilai Y_i tinggi namun mengalami kendala fisik seperti pingsan atau tekanan darah turun setelah proses donor. Kondisi tersebut menunjukkan bahwa nilai Y_i tinggi belum tentu sepenuhnya menggambarkan kesiapan fisiologis pendonor. Oleh karena itu, peneliti menerapkan *margin of error* dalam proses klasifikasi kelayakan.

Dengan mengambil ambang batas pada persentil ke-25, penelitian ini memberikan toleransi terhadap variasi individu yang mungkin memiliki nilai kelayakan sedikit di bawah rerata namun secara medis tetap aman untuk melakukan donor. Pendekatan ini juga bertujuan untuk mengurangi risiko kesalahan klasifikasi (false positive), yakni pendonor yang dianggap layak secara matematis tetapi tidak layak secara medis di lapangan.

Secara statistik, persentil ke-25 merepresentasikan kuartil bawah dari populasi nilai Y_i , yang berarti 25% data terendah dianggap tidak memenuhi standar minimum kelayakan. Dengan demikian, pendekatan ini menyeimbangkan antara akurasi sistem pengambilan keputusan dan kondisi empiris di lapangan, serta memberikan margin keamanan tambahan dalam menentukan pendonor darah yang benar-benar layak.

Penggunaan metode MOORA dalam penelitian ini relevan karena proses penentuan kelayakan pendonor darah melibatkan banyak kriteria medis yang harus dinilai secara bersamaan. MOORA mampu memberikan nilai optimasi yang mencerminkan tingkat kelayakan setiap calon pendonor secara objektif dan terukur. Dengan demikian, hasil yang diperoleh dari metode MOORA dapat dijadikan dasar yang kuat dalam memberikan rekomendasi kelayakan pendonor darah di PMI Tangerang.

Dengan karakteristik tersebut, MOORA dapat dikatakan sebagai metode pengambilan keputusan yang empiris, sistematis, logis, dan relevan untuk digunakan dalam berbagai bidang penelitian maupun implementasi nyata.

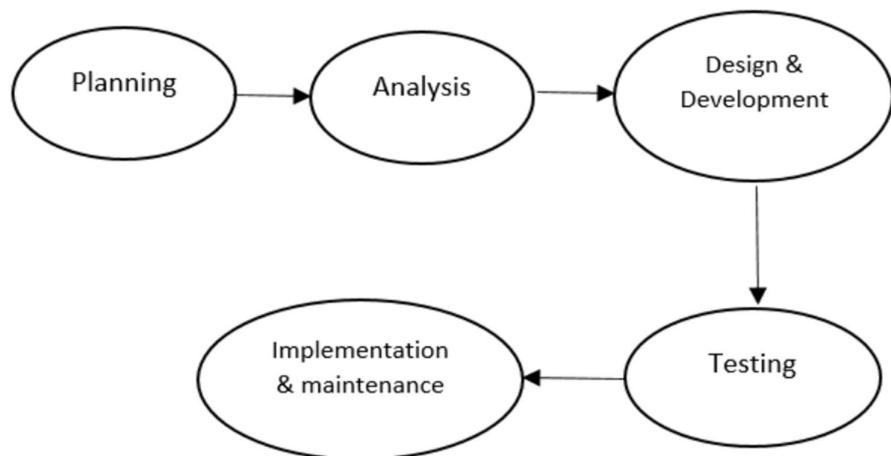
2.3 Teori Framework yang Digunakan

Framework atau kerangka kerja pada implementasi uji kelayakan pendonor menggunakan metode MOORA berbasis web ini dapat memberikan kemudahan dalam implementasi terhadap web agar penelitian ini juga dapat terstruktur. Adapun fungsi dari framework sebagai berikut[42].

- a. Penggunaan Framework dapat mempercepat proses pengembangan system
- b. Framework dapat menjaga ketetapan program
- c. Meningkatkan kualitas system.

2.3.1 Database System Development Lifecycle (DSDLC)

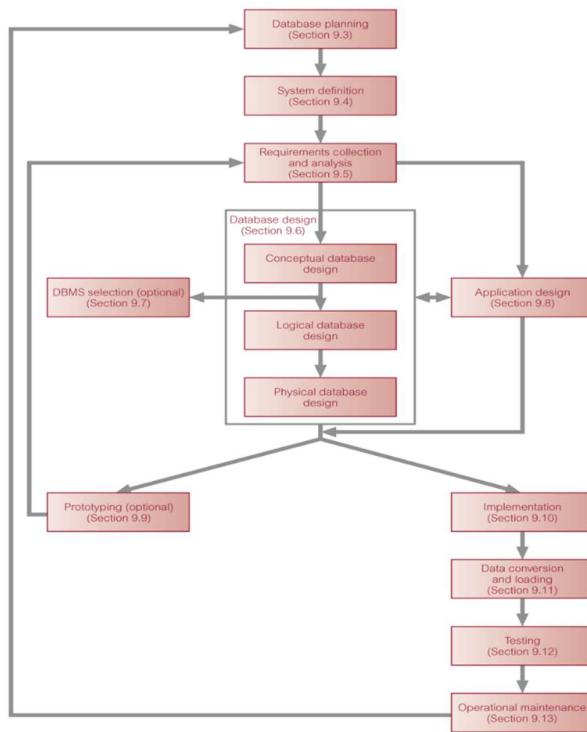
Umumnya, dalam pengembangan *Website*, pendekatan yang dilakukan untuk memudahkan manajemen progress dari pembuatan software adalah dengan *Website Development Lifecycle* (WDLC)[43]. Konsep WDLC sendiri menjadi dasar untuk model pengembangan *website*, seperti waterfall, prototyping, dan spiral . Waterfall adalah metode yang memiliki kerangka kerja terstruktur dan perlu untuk melewati beberapa tahapan untuk mendapatkan hasil akhir yang diinginkan[44]



Gambar 2. 11 Tahapan dalam Metode WDLC [43]

Selain WDLC, masih ada turunan lifecycle lain yang dapat digunakan. Ketika produk yang dikembangkan lebih difokuskan untuk pengembangan database, nama dari lifecycle tersebut berubah menjadi Database System Development Lifecycle (DSDLC) [45]. Siklus pengembangan database ini akan sangat penting terutama untuk pengembangan database skala menengah sampai database skala besar dengan jumlah pengguna mencapai lebih dari ribuan pengguna.





Gambar 2. 12 Database System Development Lifecycle (DSDLC)[46]

Berdasarkan di atas, ada delapan tahapan dalam pengembangan siklus database.

Di dalam DSDLC terdapat beberapa tahapan yang bersifat repetisi atau mengulang seperti looping. Adanya pengulangan ini didapatkan dari feedback di dalam pengembangannya, sehingga ada beberapa tahapan yang digambarkan looping untuk menunjukkan bahwa tahapan siklus ini bisa menyesuaikan dengan situasi dan kondisi dalam pengembangan[47].

Berikut adalah penjelasan dari delapan tahap siklus pengembangan database system atau DSDLC[46], [48].

1) Database Planning

Tahap ini adalah tahapan untuk merencanakan tujuan dari pengembangan database dan alur rencana pembuatan database dengan cara yang paling efektif dan efisien.

2) System Definition

Tahap kedua ini adalah tahap untuk memberikan deskripsi cakupan dan batasan dari database yang akan dikembangkan dan juga target pengguna yang akan menggunakan database.

3) Requirements Analysis

Tahapan ketiga dari proses DSDLC adalah requirements analysis. Tahap ini adalah tahapan untuk mengumpulkan requirements yang diperlukan untuk mengembangkan database. Informasi yang dikumpulkan di tahap ini umumnya akan terkumpul secara berantakan, sehingga perlu dibuat agar lebih terstruktur dengan requirement specification. Beberapa teknik yang masuk ke dalam requirement specification adalah Structured Analysis and Design (SAD), Data Flow Diagrams (DFD), dan Hierarchical Input Process Output (HIPO) charts.

4) Database Design

Tahap keempat adalah merancang database sesuai dengan informasi yang sudah diidentifikasi di tahap ketiga. Ada dua pendekatan yang sering dipakai dalam melakukan desain database, yaitu bottom-up dan top-down. Kedua pendekatan ini memiliki keunggulan dan kekurangannya tersendiri, sehingga akan lebih baik untuk menggunakan kedua pendekatan ini sesuai dengan kondisi saat akan melakukan pengembangan database agar pendekatan yang digunakan bisa lebih efektif.

5) Prototyping

Tahapan prototyping ini merupakan tahapan opsional, jadi bisa disesuaikan kebutuhannya dalam pengembangan database. Prototyping sendiri adalah tahapan untuk membangun model dari database, yang memberikan izin kepada user untuk mengevaluasi tampilan dan fungsi dari hasil akhir sistem yang akan dibuat.

1. Conceptual database design

Pembuatan desain secara konseptual merupakan tahapan pertama dari database design. Tahapan conceptual database design hanya perlu

fokus pada model data yang akan digunakan tanpa harus memikirkan logika penyimpanan database dan pertimbangan fisik database.

2. Logical database design

Database yang telah dirancang secara abstrak secara konseptual akan diubah menjadi logical design yang memiliki relasi antar tabel. Tahapan yang dilakukan di logical design umumnya adalah melakukan normalisasi database untuk melihat kesesuaian model data.

3. Physical database design

Physical design adalah tahapan yang difokuskan untuk mengimplementasikan logical database design yang sudah dibuat. Tahapan ini lebih difokuskan ke struktur database dan integritas data yang akan disimpan di database tersebut.

6) Implementation

Apabila sudah membuat desain database dan/ atau membuat prototyping, tahapan selanjutnya adalah implementasi dan penerapan rancangan desain database ke dalam bentuk database asli. Tahapan implementasi mencakup pembuatan database dan relasi antar tabel di database sesuai dengan rancangan desain.

7) Data Conversion and Loading

Tahapan selanjutnya adalah tahapan loading data untuk memasukkan data ke database yang telah dibuat. Jika terdapat data dari sistem yang lama, nantinya akan dikonversi menjadi data dengan format sistem yang baru.

8) Database Testing

Setelah memasukkan data ke dalam database, tahapan selanjutnya adalah melakukan pengujian database. Database akan diuji dan divalidasi berdasarkan requirements yang ditentukan dari users untuk memastikan proses kerja database sudah berjalan dengan baik.

9) Operational Maintenance

Tahapan terakhir dalam pengembangan database adalah operational maintenance. Tahapan ini dilakukan untuk memelihara dan memastikan database tetap berjalan dengan baik.

Dalam penelitian ini, DSDLc digunakan sebagai landasan teoritis untuk memahami tahapan perancangan basis data, sedangkan implementasinya disesuaikan dengan pendekatan bottom-up yang berfokus pada pengolahan data aktual.

2.4 Pendekatan Perancangan Database

Perancangan basis data dapat dilakukan dengan beberapa pendekatan yang bertujuan untuk menghasilkan struktur data yang sesuai dengan kebutuhan sistem. Pendekatan perancangan basis data digunakan untuk menentukan alur berpikir dalam membangun basis data, mulai dari identifikasi kebutuhan hingga implementasi dalam sistem manajemen basis data[49]. Pemilihan pendekatan perancangan basis data yang tepat sangat penting karena akan mempengaruhi struktur basis data yang dihasilkan, tingkat konsistensi data, serta kesesuaian basis data dengan proses bisnis yang dijalankan. Dua pendekatan yang umum digunakan dalam perancangan basis data adalah pendekatan top-down dan pendekatan bottom-up[50]. Dalam penelitian ini, pendekatan *bottom-up* dipilih karena data mengenai kriteria pemeriksaan pendonor darah di PMI Kota Tangerang sudah tersedia secara eksplisit dalam formulir pemeriksaan kesehatan. Dengan menggunakan pendekatan ini, perancang dapat memastikan bahwa setiap kolom data yang ada pada sistem SPK yang dibangun benar-benar mencerminkan data aktual yang digunakan oleh petugas medis di lapangan.

2.4.1 Pendekatan Bottom-Up dalam Perancangan Basis Data

Berbeda dengan *top-down*, pendekatan *bottom-up* dimulai dari tingkat atribut yang paling mendasar. Pendekatan ini bekerja dengan cara mengumpulkan seluruh atribut data yang diperoleh dari dokumen sumber,

seperti formulir pendaftaran, laporan medis, atau kuitansi yang sudah ada di organisasi. Atribut-atribut tersebut kemudian dianalisis, dikelompokkan, dan direorganisasi melalui proses normalisasi hingga terbentuk entitas-entitas yang stabil. Pendekatan bottom-up merupakan pendekatan perancangan basis data yang dimulai dari data yang sudah ada atau data yang dikumpulkan dari berbagai sumber, kemudian digabungkan dan disusun menjadi struktur basis data yang terintegrasi[50]. Pada pendekatan ini, fokus utama terletak pada pengelompokan atribut data menjadi tabel-tabel yang saling berelasi. Tahapan dalam pendekatan bottom-up meliputi:

1. Identifikasi atribut-atribut data yang tersedia.
2. Pengelompokan atribut menjadi entitas dan tabel.
3. Penentuan relasi antar tabel.
4. Normalisasi basis data untuk menghilangkan redundansi.
5. Integrasi tabel ke dalam basis data yang utuh.

Pendekatan bottom-up umumnya digunakan pada sistem yang telah memiliki data historis atau pada proses integrasi beberapa basis data yang sudah ada sebelumnya[51]. Pendekatan top-down dan bottom-up memiliki karakteristik dan tujuan yang berbeda. Pendekatan top-down lebih menekankan pada analisis kebutuhan sistem dan proses bisnis, sehingga cocok digunakan untuk sistem yang dirancang dari awal. Sementara itu, pendekatan bottom-up lebih berfokus pada pengolahan data yang sudah tersedia dan sering digunakan dalam pengembangan sistem yang bersifat integratif[49].

Dalam konteks sistem pendukung keputusan, pendekatan top-down dinilai lebih sesuai karena memungkinkan perancang sistem untuk terlebih dahulu memahami permasalahan dan kebutuhan pengambilan keputusan sebelum menentukan struktur basis data yang mendukung proses tersebut.

2.5 Data Flow Diagram

Data Flow Diagram (DFD) merupakan salah satu teknik pemodelan yang digunakan dalam pendekatan Database System Development Lifecycle (DSDLC) untuk menggambarkan alur data dan proses yang terjadi dalam suatu sistem[52], [53]. DFD digunakan untuk menunjukkan bagaimana data mengalir dari satu proses ke proses lainnya, mulai dari data masuk hingga menghasilkan keluaran yang diinginkan.

DFD berfokus pada apa yang dilakukan sistem terhadap data, bukan pada bagaimana sistem diimplementasikan secara teknis. Oleh karena itu, DFD sangat sesuai digunakan pada tahap analisis sistem untuk memahami proses bisnis dan kebutuhan data sebelum dilakukan perancangan sistem secara lebih rinci.

Dalam penelitian ini, DFD digunakan untuk memodelkan proses penentuan kelayakan pendonor darah pada PMI Kota Tangerang, sehingga alur data dan proses pengambilan keputusan dapat dipahami secara terstruktur.

Simbol yang digunakan pada diagram alir data atau data flow diagram antara lain, sebagai berikut :

1. Kesatuan luar (External Entity)

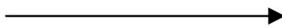
Kesatuan luar (External Entity) merupakan kesatuan di lingkungan luar sistem yang dapat berupa orang, organisasi, atau sistem lain. Lingkungan luar dan batas sistem dinotasikan dalam simbol sama



Gambar 2. 13 Simbol Kesatuan Luar

2. Arus Data (Data Flow)

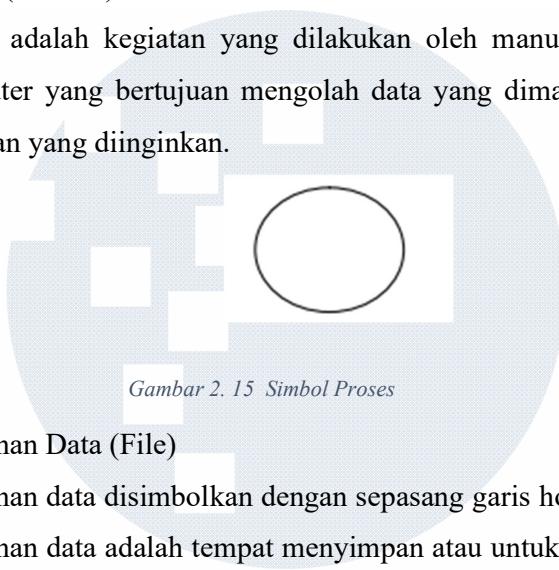
Arus data (Data Flow) di DFD diberi simbol suatu panah. Arus data ini mengalir diantara proses (process), simpanan data (data store) dan kesatuan luar (external entity).



Gambar 2. 14 Simbol Arus Data

3. Proses (Process)

Proses adalah kegiatan yang dilakukan oleh manusia, mesin, atau komputer yang bertujuan mengolah data yang dimasukkan menjadi keluaran yang diinginkan.



Gambar 2. 15 Simbol Proses

4. Simpanan Data (File)

Simpanan data disimbolkan dengan sepasang garis horizontal paralel. Simpanan data adalah tempat menyimpan atau untuk mengakses data yang diperlukan oleh sistem.



Gambar 2. 16 Simbol Simpanan Data

2.6 Teori Mengenai Tools/software yang digunakan

2.6.1 Vercel

Vercel adalah sebuah platform *cloud deployment* berbasis *serverless* yang dirancang untuk mempermudah proses pembangunan (*development*), pengujian, dan penerapan (*deployment*) *Website*[42]. Dalam penelitian ini, Vercel digunakan sebagai platform *hosting* dan *deployment* untuk aplikasi web yang dibangun

menggunakan Node.js, Express.js, dan Next.js. Vercel menyediakan lingkungan yang mendukung *runtime* JavaScript sehingga memungkinkan aplikasi berjalan dengan performa tinggi, skalabilitas otomatis (*auto-scaling*), serta pengelolaan server yang minimal. Penggunaan Vercel membantu memastikan bahwa aplikasi dapat diakses secara publik, memiliki waktu respons cepat, dan dapat di-deploy secara berkelanjutan dari repositori Git seperti GitHub.

2.6.2 Software Pengembangan Sistem

Dalam pembangunan sistem pengelolaan hak akses pengguna, diperlukan perangkat lunak pendukung yang berfungsi sebagai sarana dalam menulis, menguji, dan mengelola kode program berbasis web.

Pada tahap implementasi, penelitian ini menggunakan Visual Studio Code (VS Code) sebagai *Integrated Development Environment (IDE)*. VS Code dipilih karena memiliki dukungan lintas platform, antarmuka yang ringan, serta ketersediaan berbagai ekstensi yang dapat meningkatkan produktivitas pengembang, seperti fitur *syntax highlighting*, *debugging*, dan integrasi dengan *version control system* (Git).

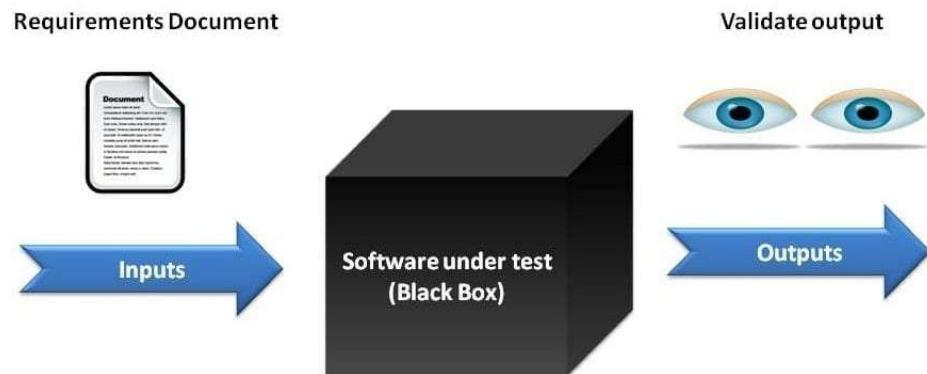
2.6.3 Server

Untuk manajemen database web, penelitian ini menggunakan Neon. Neon merupakan layanan *Backend as a Service (BaaS)* yang berbasis PostgreSQL, menyediakan fitur autentikasi, penyimpanan data, hingga API yang dapat langsung diintegrasikan dengan aplikasi web. Pemilihan Neon didasarkan pada fleksibilitas, skalabilitas, serta kemudahan integrasi yang sesuai dengan kebutuhan sistem pengelolaan hak akses pengguna.

2.7 User Acceptance Test: Black Box Testing

Black Box Testing atau pengujian fungsionalitas merupakan teknik pengujian fungsionalitas software yang merancang kasus uji berdasarkan informasi dan

spesifikasi dengan mengharapkan suatu output yang diinginkan dari setiap nilai input yang dimasukkan[54].



Gambar 2. 17 Black Box Testing

Adapun kelebihan dari black box testing diantaranya :

1. Pengujian dilakukan pada syarat – syarat dari sudut pandang user pelanggan.
2. Karena pengujian ini tidak memerlukan keterampilan dalam pemrograman, penguji tidak harus menjadi seseorang yang ahli programming dan sebagainya.
3. Pengujian yang efisien saat digunakan pada sistem yang besar.

Adapun kelemahan dari black box testing, diantaranya:

1. Dapat menimbulkan pengujian yang redundant jika developer sudah melakukan pengetesan terhadap suatu fungsionalitas atau kasus uji.
2. Merancang kasus uji bisa sulit jika persyaratannya tidak jelas dan ringkas.
3. Tidak efisien jika menguji kode yang kompleks.
4. Hasil tes yang sering dilebih-lebihkan.

2.7.2 Tujuan Pengujian

Pengujian sistem dilakukan untuk memastikan bahwa Sistem Pendukung Keputusan (SPK) Kelayakan Pendonor Darah berbasis web yang dikembangkan telah berjalan sesuai dengan kebutuhan fungsional dan non-fungsional. Pengujian ini juga bertujuan untuk memvalidasi implementasi metode *Multi-Objective Optimization on the Basis of Ratio Analysis* (MOORA) dalam menghasilkan rekomendasi kelayakan pendonor darah secara objektif, konsisten, dan akurat.

Secara khusus, pengujian sistem bertujuan untuk:

1. Memastikan setiap fitur sistem berjalan sesuai dengan spesifikasi yang telah dirancang.
2. Menilai kesesuaian hasil perhitungan MOORA antara perhitungan sistem dan perhitungan manual.
3. Menguji kemudahan penggunaan sistem oleh petugas PMI.
4. Memastikan sistem mampu memproses data pendonor secara cepat dan stabil.

2.7.3 Metode Pengujian

Metode pengujian yang digunakan dalam penelitian ini meliputi:

- 1) Pengujian Fungsional (Black Box Testing)
Pengujian ini berfokus pada fungsi utama sistem tanpa melihat struktur kode program. Setiap fitur diuji berdasarkan input yang diberikan dan output yang dihasilkan.
- 2) Pengujian Validasi Perhitungan MOORA
Pengujian ini dilakukan dengan membandingkan hasil perhitungan MOORA yang dihasilkan oleh sistem dengan hasil perhitungan manual menggunakan data sampel pendonor.
- 3) Pengujian Usability (Kegunaan Sistem)

Pengujian ini bertujuan untuk mengetahui tingkat kemudahan penggunaan sistem oleh pengguna akhir, yaitu petugas PMI.

2.8 Donor Darah

2.8.1 Definisi

Pendonor darah merupakan individu yang melakukan pemberian komponen darahnya, seperti plasma, trombosit, maupun sel darah merah, untuk digunakan dalam proses transfusi kepada pasien yang membutuhkan. Tujuan utama dari donor darah adalah mendukung proses penyembuhan penyakit, pemulihan kesehatan, serta penyelamatan nyawa pada kondisi darurat medis, seperti kecelakaan, operasi besar, atau pasien dengan penyakit kronis tertentu. Sebelum proses pengambilan darah dilakukan, setiap calon pendonor wajib melalui serangkaian pemeriksaan kesehatan untuk memastikan bahwa darah yang akan didonorkan aman, baik bagi penerima maupun bagi pendonor itu sendiri. Pemeriksaan tersebut umumnya meliputi pengecekan tekanan darah, kadar hemoglobin, berat badan, usia, riwayat penyakit, konsumsi obat, serta gaya hidup yang berkaitan dengan kesehatan. Selain itu, proses donor darah juga harus didasarkan pada kesukarelaan dan persetujuan penuh dari pendonor, sehingga aspek etika dan hak individu tetap terjamin [55].

Donor darah merupakan salah satu bentuk kepedulian sosial yang seharusnya menjadi tanggung jawab setiap individu dalam masyarakat untuk membantu sesama. Namun, masih banyak masyarakat yang belum memahami manfaat donor darah bagi kesehatan, bahkan sebagian merasa enggan melakukannya karena adanya kekhawatiran terhadap kemungkinan efek samping. Padahal, aktivitas donor darah justru memberikan dampak positif, salah satunya adalah merangsang pembentukan sel darah baru sehingga sirkulasi darah dalam tubuh tetap optimal. Jika dilakukan secara rutin setiap tiga bulan sekali, donor darah dapat membantu menjaga kondisi tubuh agar tetap sehat sekaligus

memberikan manfaat besar bagi orang lain yang membutuhkan transfusi darah. Dengan demikian, donor darah tidak hanya bernilai sosial, tetapi juga memberikan keuntungan dari sisi kesehatan bagi pendonor itu sendiri.

2.8.2 Jenis Golongan Darah

Golongan darah manusia dibedakan berdasarkan sistem ABO dan Rhesus (Rh). Berdasarkan sistem ABO, darah dikelompokkan menjadi golongan A, B, AB, dan O, sedangkan berdasarkan faktor Rhesus dibedakan menjadi Rh positif dan Rh negatif. Setiap golongan darah memiliki karakteristik antibodi dan antigen yang berbeda sehingga menentukan kecocokan dalam proses transfusi darah. Ketidaksesuaian golongan darah dapat menimbulkan reaksi transfusi yang berbahaya bagi penerima, mulai dari reaksi ringan hingga reaksi hemolitik yang dapat berakibat fatal.

2.8.3 Pendonor

Menurut *World Health Organisation* (WHO), pendonor darah di klasifikasikan menjadi 3 tipe yaitu, donor sukarela yang tidak dibayar, donor keluarga atau pengganti dan donor komersial atau professional[56]

a. Pendonor Sukarela

Pendonor Sukarela merupakan Individu yang menyumbangkan darah, plasma, atau komponen darah atas inisiatif sendiri. Mereka tidak menerima kompensasi finansial, baik dalam bentuk tunai maupun barang yang setara dengan uang. Bentuk apresiasi berikut ini umumnya dikecualikan dari kategori pembayaran atau pengganti uang:

- i. Penghargaan berupa lencana atau sertifikat kehormatan yang tidak memiliki nilai ekonomi.
- ii. Penggantian biaya transportasi yang secara spesifik dikeluarkan untuk keperluan mendonorkan darah.

- iii. Penyediaan makanan ringan sebelum, selama, atau setelah proses donor darah.

b. Donor keluarga atau pengganti

Individu yang menyumbangkan darah ketika anggota keluarga, kenalan, atau komunitas mereka memerlukan. Dalam mekanisme ini, darah yang diperlukan oleh pasien diperoleh dari satu atau beberapa penyumbang dari dalam keluarga pasien atau komunitas. Sering kali, staf rumah sakit meminta keluarga pasien untuk menyumbangkan darah. Namun, dalam beberapa situasi, merupakan kewajiban bagi setiap pasien yang memerlukan transfusi untuk menyediakan sejumlah darah dari penyumbang pengganti saat pendaftaran, dalam keadaan darurat, atau sebelum prosedur bedah yang telah dijadwalkan. Pada dasarnya, mereka tidak menerima kompensasi finansial untuk layanan transfusi darah atau bank darah rumah sakit, tetapi terkadang mereka menerima imbalan uang atau bentuk apresiasi lain dari keluarga pasien sebagai tanda terima kasih.

c. Donor komersial atau Professional

Donor komersial atau profesional adalah individu yang menerima pembayaran berupa uang atau bentuk imbalan lain atas darah yang mereka donasikan. Mereka biasanya mendonorkan darah secara berkala dan dapat memiliki perjanjian dengan bank darah untuk menyuplai darah dengan imbalan yang telah ditentukan. Motivasi utama donor komersial atau profesional adalah imbalan yang mereka terima, bukan keinginan untuk menolong sesama.

2.8.4 Resiko Donor Darah

Kegiatan donor darah memiliki risiko medis apabila pendonor tidak memenuhi syarat kelayakan. Risiko yang sering terjadi antara lain pusing, lemas, mual, penurunan tekanan darah secara drastis, hingga pingsan setelah

proses donor. Pada kasus tertentu, donor yang dilakukan pada individu dengan kondisi kesehatan yang tidak layak juga dapat menyebabkan komplikasi lebih lanjut seperti gangguan peredaran darah. Selain itu, darah yang berasal dari pendonor yang tidak layak dapat menurunkan kualitas darah dan berisiko menimbulkan efek buruk bagi pasien penerima transfusi.

Proses donor darah menyebabkan berkurangnya volume darah sementara dalam tubuh pendonor. Secara fisiologis, tubuh akan merespons dengan meningkatkan produksi sel darah merah untuk menggantikan darah yang hilang. Pada individu yang sehat, proses ini tidak menimbulkan masalah serius. Namun, pada individu dengan kadar hemoglobin rendah, tekanan darah tidak stabil, atau kondisi kesehatan tertentu, donor darah dapat menimbulkan gangguan seperti kelelahan berlebihan, penurunan daya tahan tubuh sementara, dan gangguan sirkulasi darah. Oleh karena itu, pemeriksaan kondisi fisiologis pendonor sebelum melakukan donor darah merupakan langkah yang sangat penting. Setiap kriteria dalam penentuan kelayakan pendonor memiliki hubungan langsung dengan keselamatan pendonor dan kualitas darah yang dihasilkan. Usia dan berat badan berkaitan dengan daya tahan tubuh terhadap kehilangan volume darah. Tekanan darah dan kadar hemoglobin mencerminkan kesiapan sistem peredaran darah dalam tubuh untuk menjalani proses donor. Riwayat penyakit berfungsi sebagai penyaring utama untuk mencegah penularan penyakit melalui transfusi dan menghindari risiko medis terhadap pendonor. Oleh karena itu, kesalahan dalam menilai satu saja dari kriteria tersebut dapat berdampak langsung terhadap keselamatan pendonor dan pasien penerima darah, sehingga proses seleksi kelayakan harus dilakukan secara teliti, objektif, dan berbasis data.

2.8.5 Syarat Pendonor Darah

Untuk pendonor yang ingin mendonorkan darahnya, ada beberapa syarat yang harus dipenuhi oleh calon pendonor. Berikut syarat yang harus dipenuhi.

- a. Usia antara 17 sampai 60 tahun (individu berusia 17 tahun dapat menjadi donor dengan surat persetujuan dari orang tua. Hingga usia 60 tahun, donor masih dapat menyumbangkan darahnya dengan jeda waktu tiga bulan berdasarkan evaluasi medis).
- b. Bobot tubuh minimal 45 kg.
- c. Suhu badan: 36,6 - 37,5°C (oral).
- d. Tekanan darah stabil, dengan nilai sistolik berkisar 110 - 160 mm Hg dan diastolik antara 70 - 100 mm Hg.
- e. Detak jantung teratur dengan frekuensi 50 - 100 kali per menit.
- f. Tingkat Hemoglobin untuk wanita tidak kurang dari 12,0 gram%, dan untuk pria tidak kurang dari 12,5 gram%.
- g. Jumlah penyumbangan dalam setahun paling banyak 4 kali dengan jarak penyumbangan sekurang kurangnya 3 bulan.

