

BAB II

LANDASAN TEORI

2.1 Sepakbola

Sepakbola adalah olahraga yang dimainkan dengan sebuah bola antara 2 tim yang berisi 11 pemain. Tujuan utama dari sepakbola adalah mencetak gol lebih banyak dari tim musuh. Sebuah gol adalah ketika bola yang ditendang seorang pemain masuk kedalam gawang musuh.

1 tim sepakbola terdiri dari 11 pemain. Salah satu pemain tersebut adalah penjaga gawang, yang memiliki tugas untuk menjaga agar bola tidak masuk kedalam gawang. Posisi tersebut adalah salah satu yang diperbolehkan menyentuh bola dengan tangan. 10 pemain lain dibagi dalam 3 kategori, yaitu defender yang memiliki tugas utama membantu pertahanan, forward yang memiliki tugas utama menyerang, dan midfielder yang memiliki tugas kombinasi dari 2 posisi tersebut.

[13]

2.2 Database

Database adalah kumpulan data yang disimpan dalam sebuah komputer. Data dalam sebuah database disimpan dalam bentuk *records*, dimana penerapannya dapat berupa *medical records*, *employee records* dll.

Database terdiri dari 4 komponen, yaitu [14]:

- a. Data Item: Unit terkecil dalam sebuah database. Dalam contoh buku telepon, nama merupakan salah satu contoh Data Item.
- b. Relationships: Hubungan dari berbagai elemen database.
- c. Constraints: Aturan-aturan yang ditetapkan dalam database yang benar.
- d. Schema: Menjelaskan tentang organisasi data dan *relationships* dalam database tersebut.

Database memiliki beberapa ciri utama, yaitu [14]:

- a. Database bersifat tetap, untuk menghindari perubahan atau kehilangan data yang tidak disengaja.
- b. Database dapat digunakan lebih dari satu orang.
- c. Database harus menunjukkan data yang akurat sesuai hasil *input*.
- d. Database dapat menampung banyak *records*.
- e. Database dapat digunakan dan diandalkan.

2.3 HTML

HTML (Hypertext Markup Language) adalah bahasa yang digunakan untuk membuat struktur dalam sebuah halaman web. Semua dokumen HTML dimulai dengan *tag* `<html>` dan diakhiri oleh *tag* penutup yaitu `</html>`. *Tag* dasar lainnya, yaitu `<head>` dan `<body>` merupakan bagian utama dari sebuah dokumen HTML. `<head>` akan berisi informasi tentang halaman tersebut, seperti `<link>`, `<title>`, `<meta>`, `<style>` dll. *Tag* `<body>` berada dibawah `<head>`, dan merupakan isi dari halaman web tersebut. HTML memiliki perbedaan dibanding bahasa pemrograman lainnya, dikarenakan HTML hanya memberi instruksi dalam menampilkan konten dalam sebuah *browser*. [15]

2.4 PHP

PHP (Pre-processor Hypertext) adalah *server-side scripting language* yang didesain secara spesifik untuk pembuatan website. Kode PHP dapat disisipkan kedalam sebuah halaman HTML dan akan dijalankan setiap halaman tersebut dibuka. Kode PHP dimulai dengan simbol `<?php` dan diakhiri dengan `?>`. *Tags* PHP tersebut menandakan ke server website bahwa kode didalam *Tag* tersebut akan diproses sebagai PHP, sedangkan semua kode diluar *tag* tersebut akan diproses sebagai HTML. Salah satu fungsi dari PHP adalah untuk menghubungkan halaman web tersebut dengan sebuah database. [16]

2.5 PHPMyAdmin

PHPMyAdmin adalah sebuah aplikasi yang ditulis dalam bahasa PHP dan berisi XHTML, CSS dan JavaScript client code. PHPMyAdmin menyediakan

interface website yang lengkap untuk administrasi MySQL. Tujuan utama dari PHPMyAdmin adalah untuk memberi *web-based management* dari server dan data MySQL. PHPMyAdmin memiliki berbagai fitur seperti *browser-based management, privileges setting (Administrator), real-time interface* dll. [17]

2.6 MySQL

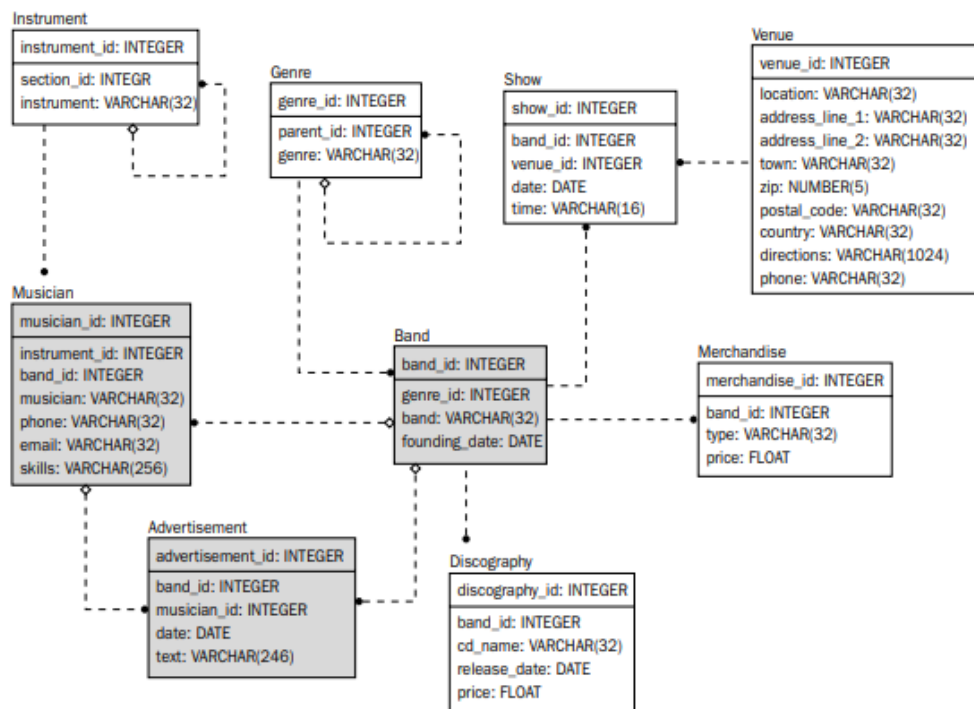
Server MySQL mengontrol akses ke data dan memastikan banyak pengguna dapat mengakses secara bersamaan. MySQL juga menjamin kecepatan dan keamanan dalam mengakses data tersebut. Server MySQL menggunakan bahasa SQL (Standard Query Language), yang merupakan sebuah bahasa yang digunakan untuk mengakses *relational database management systems (RDBMS)*. Bahasa tersebut digunakan untuk menyimpan dan mengambil data dari sebuah database. SQL juga digunakan pada sistem-sistem lain seperti Oracle, PostgreSQL, Sybase, Microsoft SQL Server dll. [17]

2.7 Laravel

Laravel adalah sebuah framework PHP yang digunakan untuk membangun sebuah sistem berbasis website. Laravel digunakan untuk melengkapi dan mendukung seorang *developer*. Framework tersebut memiliki tujuan untuk menyediakan sekumpulan *code* yang sederhana dan berbagai fitur yang membantu *developer* mempelajari dan membuat sistem dengan mudah. [18]

2.8 Entity Relationship Diagram

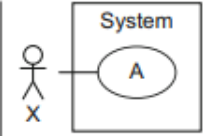

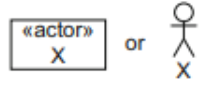
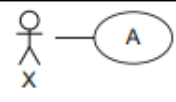
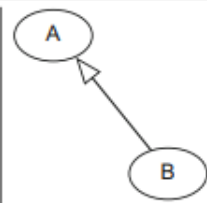
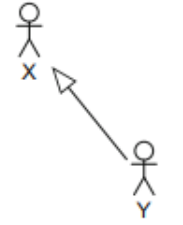
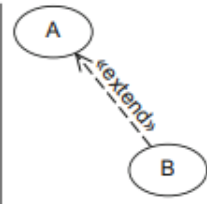
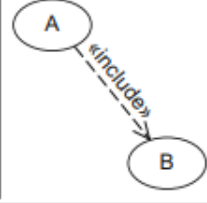
Entity Relationship Diagram (ERD) menggambarkan struktur logis dari sebuah database. Berikut adalah contoh simbol-simbol dari sebuah diagram ERD dalam Gambar 2.1.



Gambar 2.1 Contoh Entity Relationship Diagram [19]

Keuntungan dalam penggunaan Entity Relationship Diagram adalah penjelasan elemen-elemen dalam database seperti entitas, *relationship*, atribut dan *cardinality*, dalam bentuk gambar yang mudah dimengerti. [20]







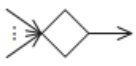


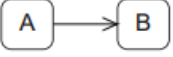
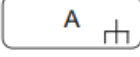
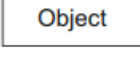
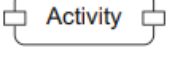
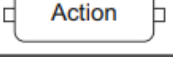
2.9 Use Case Diagram

<i>Name</i>	<i>Notation</i>	<i>Description</i>
System		Boundaries between the system and the users of the system
Use case		Unit of functionality of the system
Actor		Role of the users of the system
Association		X participates in the execution of A
Generalization (use case)		B inherits all properties and the entire behavior of A
Generalization (actor)		Y inherits from X; Y participates in all use cases in which X participates
Extend relationship		B extends A: optional incorporation of use case B into use case A
Include relationship		A includes B: required incorporation of use case B into use case A

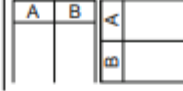

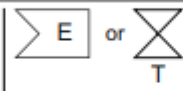
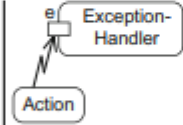
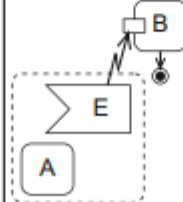
Gambar 2.2. Tabel Notasi Use Case Diagram

Use Case Diagram membantu mendeskripsikan berbagai skenario dalam menggunakan sebuah sistem. Diagram tersebut membantu mengekspresikan berbagai hal yang sebuah sistem harus lakukan tanpa penjelasan mengenai *data structures*, algoritma dll. Gambar 2.2 menjelaskan mengenai berbagai notasi dalam pembuatan *Use Case Diagram*. [21]

2.10 Activity Diagram

Name	Notation	Description
Action node		Actions are atomic, i.e., they cannot be broken down further
Activity node		Activities can be broken down further
Initial node		Start of the execution of an activity
Activity final node		End of ALL execution paths of an activity
Flow final node		End of ONE execution path of an activity
Decision node		Splitting of one execution path into two or more alternative execution paths
Merge node		Merging of two or more alternative execution paths into one execution path
Parallelization node		Splitting of one execution path into two or more concurrent execution paths
Synchronization node		Merging of two or more concurrent execution paths into one execution path
Edge		Connection between the nodes of an activity
Call behavior action		Action A refers to an activity of the same name
Object node		Contains data and objects that are created, changed, and read
Parameters for activities		Contain data and objects as input and output parameters
Parameters for actions (pins)		Contain data and objects as input and output parameters

Gambar 2.3. Tabel Notasi Activity Diagram

<i>Name</i>	<i>Notation</i>	<i>Description</i>
Partition		Grouping of nodes and edges within an activity
Send signal action		Transmission of a signal to a receiver
Asynchronous accept (time) event action		Wait for an event E or a time event T
Exception handler		Exception handler is executed instead of the action in the event of an error θ
Interruptible activity region		Flow continues on a different path if event E is detected

Gambar 2.4. Tabel Notasi Activity Diagram (lanjutan)

Dalam perancangan sistem, activity diagram digunakan untuk menggambarkan berbagai aktivitas, objek, event dan state dalam sebuah sistem. Tujuan dari pembuatan diagram tersebut adalah menggambarkan perilaku sebuah sistem untuk berbagai aktivitas. [22] Gambar 2.3, 2.4 merupakan contoh dari activity diagram. [21]

2.11 Class Diagram

Name	Notation	Description
Class		Description of the structure and behavior of a set of objects
Abstract class		Class that cannot be instantiated
Association		Relationship between classes: navigability unspecified (a), navigable in both directions (b), not navigable in one direction (c)
N-ary association		Relationship between N (in this case 3) classes
Association class		More detailed description of an association
xor relationship		An object of A is in a relationship with an object of B or with an object of C but not with both
Strong aggregation = composition		Existence-dependent parts-whole relationship (A is part of B ; if B is deleted, related instances of A are also deleted)
Shared aggregation		Parts-whole relationship (A is part of B ; if B is deleted, related instances of A need not be deleted)
Generalization		Inheritance relationship (A inherits from B)
Object		Instance of a class
Link		Relationship between objects

Gambar 2.5. Tabel Notasi Class Diagram

Class Diagram digunakan untuk membuat model struktur statis sebuah sistem. Diagram tersebut menjelaskan mengenai berbagai elemen dalam sebuah

sistem dan relasi diantara mereka. Gambar 2.5 menjelaskan mengenai notasi-notasi dalam pembuatan *Class Diagram*. [21]

2.12 Waterfall Model

Model waterfall adalah pendekatan tradisional untuk *software engineering*. Saat menggunakan metode waterfall, diasumsikan bahwa seluruh informasi mengenai *system requirement* dapat diketahui sebelum *development* dimulai. Karakteristik lain dalam metode waterfall adalah proses selanjutnya hanya bisa dimulai setelah proses sebelumnya selesai. [23]

Berikut adalah tahap-tahap dalam proses pengembangan menggunakan metode waterfall. [24]

- a. Requirement: Informasi mengenai kebutuhan sistem seperti fitur-fitur yang akan dibuat.
- b. Design: Merupakan tahap *planning* dan *problem solving* berdasarkan informasi yang dikumpulkan di tahap sebelumnya.
- c. Implementation: *Requirement* dari tahap sebelumnya dirubah kedalam *production environment*
- d. Testing: Fase ini akan melakukan pengujian untuk memastikan kualitas dan kesesuaian dengan *system requirement*.
- e. Maintenance: Tahap dimana dilakukannya beberapa modifikasi dan penyesuaian setelah *software* dirilis.

2.13 Spiral Model

Model Spiral adalah proses *development* sistem yang menggabungkan berbagai elemen desain dan *prototyping* dalam konsep *top down* dan *bottom up*. Model Spiral fokus terhadap *risk assessment* dan meminimalisir resiko-resiko projek. [24]

Berikut adalah tahapan-tahapan dalam model Spiral. [24]

- a. *Planning*: Mengetahui informasi mengenai *system requirements*.
- b. *Risk Analysis*: Mengetahui resiko dan mencari solusi alternatif.
- c. *Development*: Fase pembuatan *software*.
- d. *Evaluation*: Fase evaluasi oleh customer.

2.14 Incremental dan Iterative Model

Model tersebut menggabungkan model Waterfall dalam konsep *iterative*. Setiap *release* dari sebuah proses *development*, akan menambahkan *function* hingga semua *function* yang ditetapkan dalam *system requirement* terpenuhi. [24]

2.15 User Interface / User Experience

Usability (Kegunaan) merupakan tujuan utama dari User Experience dimana pengguna dapat dengan mudah dan nyaman menggunakan sistem tersebut [25]. Untuk menjamin kenyamanan pengguna dalam menggunakan sistem, proses desain *user interface* akan mengacu kepada beberapa prinsip menurut Rosenweig (2015).

- a. *Easy to use*: Fitur-fitur mudah ditemukan dan digunakan.
- b. *Easy to understand*: Pengguna tidak memerlukan bantuan saat menggunakan sistem.

- c. Visual Focus.
- d. *Clarity*: Informasi mudah dimengerti.
- e. *Effective*: Pengguna dapat menggunakan sistem secara efisien.
- f. *Works for the user*.

2.16 Simple Additive Weighting Method (Decision Support)

Simple Additive Weighting (SAW) adalah suatu metode *decision support*. Simple Additive Weighting memiliki keunggulan dimana metode tersebut merupakan metode yang paling sering dan mudah digunakan. Konsep dasar dari metode Simple Additive Weighting adalah mencari jumlah bobot dari *performance rating* pada semua atribut. [26]

Berikut adalah rumus yang akan digunakan dalam algoritma SAW. [10]

2.14.1 Rumus Normalisasi

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\max_i x_{ij}}; & \text{jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{\min_j x_{ij}}{x_{ij}}; & \text{jika } j \text{ adalah atribut biaya (cost)} \end{cases}$$

Rumus 2.1 Rumus Normalisasi SAW

Keterangan:

- R_{ij} = Nilai rating kinerja ternormalisasi.
- X_{ij} = Nilai atribut yang dimiliki dari setiap kriteria.

2.14.2 Rumus Value of Preference

$$V_i = \sum_{j=1}^n W_j r_{ij}$$

Rumus 2.2. Rumus Value of Preference SAW

Keterangan:

- V_i = Ranking untuk setiap alternatif.
- W_j = Nilai bobot dari setiap kriteria.
- R_{ij} = Nilai ranting kerja ternormalisasi.

2.17 Weighted Product

Weighted Product adalah sebuah metode yang menggunakan *multiplication* kepada setiap atribut, dimana setiap rating dari masing-masing atribut harus disesuaikan dengan bobot masing-masing. Proses tersebut memiliki kemiripan dengan proses normalisasi. [26]

Berikut adalah rumus untuk menghitung *value* dalam *Weighted Product*. [26]

$$V_i = \frac{\prod_{j=1}^n X_{ij} W_j}{\prod_{j=1}^n (X_j) W_j}; i = 1, 2, \dots, m$$

Rumus 2.3. Rumus *Weighted Product*

Keterangan:

V_i : *Value* dari *alternative preferences*

X_{ij} : *Alternated attribute value*

W_j : Nilai dari *criterion weight*

2.18 ELECTRE

ELECTRE adalah salah satu metode pengambilan keputusan yang menggunakan konsep perbandingan *pairwise* berdasarkan masing-masing *criterion*. Metode ELECTRE digunakan untuk berbagai kasus dimana terdapatnya banyak alternatif, namun dengan kriteria yang sedikit. [27]

Berikut adalah rumus untuk menghitung matrix menggunakan metode ELECTRE. [27]

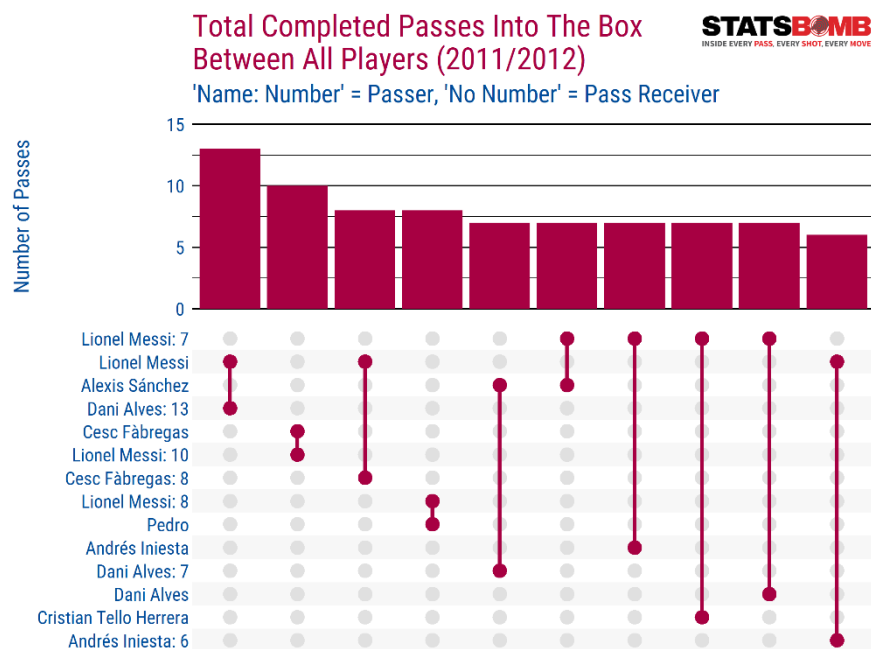
$$k_l = \frac{\max\{|V_{kj} - V_{lj}|\} j \in d_{kl}}{\max\{|V_{kj} - V_{lj}|\} \forall j}$$

Rumus 2.4. Rumus Matrix ELECTRE

2.19 Visualisasi Data

Visualisasi data memiliki tujuan untuk mengartikan data yang telah diolah menjadi bentuk gambar. Berikut adalah prinsip-prinsip dalam memvisualisasikan data [28]:

- Mengerti konteks data.
- Memilih tipe visualisasi yang cocok.
- Rapih.
- Fokus pada unsur yang ingin di visualisasi.
- Berpikir seperti desainer.
- Dapat menceritakan data dengan jelas.

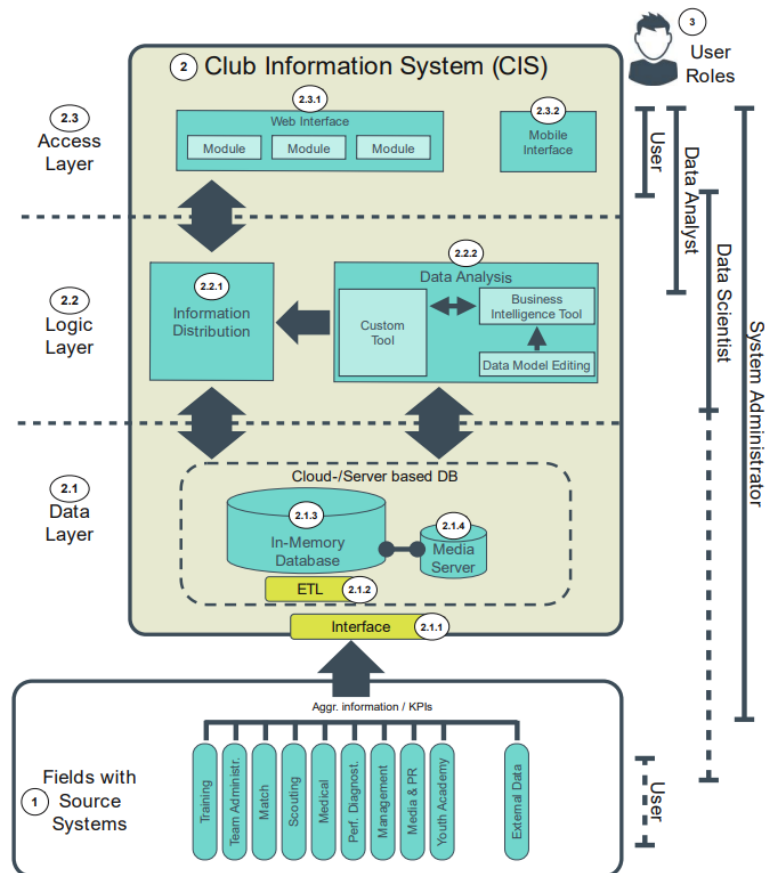


Gambar 2.6 Contoh visualisasi data pemain sepakbola.

2.20 Konsep Club Information Systems (CIS)

Dalam tim sepakbola professional, keperluan untuk mempunyai sebuah sistem informasi meningkat seiring berkembangnya teknologi. Sebuah sistem informasi dalam tim sepakbola dapat menyediakan database untuk mengelola *data science*, seperti *advanced analytics*, *visual analytics* atau *machine learning* sesuai dengan konteks yang diperlukan oleh tim tersebut. Hal tersebut dapat memberi wawasan baru dan mengungkap pola atau korelasi berbagai data yang belum yang sebelumnya belum diketahui. [29]

Pengembangan Sistem Informasi dalam tim sepakbola dapat mengacu kepada sebuah *model driven architecture* yang telah dibuat oleh Blobel & Lames (2020). Model driven architecture tersebut dibuat berdasarkan *three-layer architecture* untuk sistem BI, yang disesuaikan dengan keperluan tim sepakbola.



Gambar 2.7 Three-layer architecture untuk tim sepakbola oleh Blobel & James (2020)

Model driven architecture (Gambar 2.7) membagi sistem informasi menjadi 3 bagian, yaitu.

a. *Data Layer*

Merupakan sebuah tahap yang berfungsi untuk mengumpulkan berbagai data untuk diolah dalam sistem informasi. Hal tersebut dapat berbentuk teknologi, seperti sebuah alat yang melakukan *tracking* gerakan pemain dalam sebuah pertandingan, atau berbagai observasi dari pelatih yang diterjemahkan menjadi

sebuah angka. Angka yang berasal dari observasi seorang pelatih merupakan suatu metrik, seperti *successful tackles percentage*, *shot conversion*, *shot accuracy* dll.

b. *Logic Layer*

Logic Layer merupakan tahapan analisa data-data yang telah dikumpulkan di *Data Layer*. Tahapan ini akan menjawab berbagai system requirement yang ditetapkan diawal tahap *development website*.

c. *Access Layer*

Access layer adalah tahap dimana hasil proses dari *logic layer* akan ditampilkan ke *user* melalui sebuah *interface*.

2.21 Organizational Requirement & System Requirement CIS

Berdasarkan penelitian oleh Blobel & Lames (2020), tabel 2.1 menjelaskan tentang permasalahan yang dihadapi oleh tim sepakbola terkait pengelolaan data dan informasi, dan solusi yang diberikan dengan adanya penerapan sistem informasi.

Tabel 2.1. Organizational Requirement dan System Requirement.

Organizational Requirement	Solusi (System Requirement)
Data terpisah dari berbagai departemen.	Menggabungkan data dari berbagai sumber dalam satu sistem.
Import Data.	User interface yang membantu <i>input</i> data.
Kemudahan dalam mengakses data.	Sistem dibuat dalam bentuk website yang dapat diakses dari berbagai platform.
Security.	Sistem dibuat menggunakan platform database yang dapat

	diandalkan, dan juga terdapat fitur authorization (login, logout).
Analisa Data.	Data akan diolah menjadi berbagai metrik yang mudah dimengerti.
Rekomendasi (Decision Support).	Data yang telah diolah akan diproses dalam menggunakan metode SAW untuk membantu dalam pengambilan keputusan.
Visualisasi Data.	Data yang telah diolah divisualiasi dalam bentuk yang mudah dimengerti.

2.22 Functional Requirement Tim Sepakbola

Berdasarkan jurnal berjudul “Functional Design of Football Club Information System Using Idef Methods” oleh Zoran Nikolic, Goran Jakovljevic dan Milan Nikolic, berikut adalah functional requirement sistem yang terbagi dalam 3 bagian. [30]

Tabel 2.2. Tabel Functional Requirement

Bidang	Fitur
Analysis	Sistem pendukung keputusan menggunakan metode <i>Simple Additive Weighting</i> untuk mencari posisi terbaik seorang pemain berdasarkan atribut-atribut.
	Fitur untuk memasukkan bobot <i>custom</i> sesuai preferensi <i>user</i> .
Matches	Mengelola data mengenai <i>opponents</i> , seperti nama tim lawan dan logo.
	Mengelola data mengenai <i>fixtures</i> , seperti nama tim lawan, tanggal pertandingan dan skor akhir.
<i>Game Statistics</i>	Mengelola data mengenai <i>players</i> , seperti nama pemain, foto dan atribut.
	Mengelola data mengenai <i>player stats</i> , yaitu statistik seorang pemain dalam pertandingan.

2.23 Indikator Performa masing-masing Posisi Sepakbola

Posisi dalam sepakbola terdiri dari 4 kategori dasar, yaitu kiper, defender, midfielder dan forward. Berikut adalah daftar posisi pemain yang akan digunakan dalam pembangunan sistem rekomendasi posisi: [12]

- a. Goalkeeper
- b. Centre Back
- c. Defensive Midfield
- d. Attacking Midfield
- e. Winger
- f. Striker

Berikut adalah kriteria-kriteria yang digunakan dalam sistem decision support menggunakan algoritma Simple Additive Weighting. [12]

1. Physical

- a. *Speed*
- b. *Balance*
- c. *Strength*
- d. *Durability*
- e. *Agility*
- f. *Power*
- g. *Stamina*
- h. *Jumping*

2. Technical

- a. *Passing*
- b. *Control*
- c. *Long Pass*
- d. *Shot Accuracy*
- e. *Heading*
- f. *Tackling*
- g. *Catching*
- h. *Reflex*

3. Tactical

- a. *Positioning*
- b. *Creative*

c. *Determination*

d. *Reading the Game*

2.24 Manajemen Klub Sepakbola

Manager tim sepakbola, yang biasa dipilih oleh pemilik atau Director of Football tim tersebut, memiliki tujuan menjadi pemimpin dan membuat berbagai keputusan sebuah tim. Di era modern, manager sebuah tim fokus dalam keperluan dalam lapangan seperti melatih pemain, pemilihan taktik dan pemilihan pemain untuk sebuah pertandingan, sedangkan Director of Football fokus kepada keperluan luar lapangan seperti jual beli pemain dan finansial tim.

Berikut adalah tugas utama dari seorang manager tim sepakbola [31]:

- a. Pemilihan tim untuk sebuah pertandingan.
- b. Taktik dan cara tim tersebut bermain.
- c. Melakukan pelatihan tim dan individu.
- d. Menjaga disiplin dan kesehatan pemain.
- e. Berbicara dengan pers mengenai kondisi tim.
- f. Memimpin staff seperti asisten dan pelatih.

2.25 Statistika dalam Sepakbola

Peran statistika dalam sepakbola semakin meningkat. Matematika memiliki peran dalam menganalisa performa seorang pemain sepakbola, hingga performa seluruh tim. Berikut adalah statistika dasar yang telah digunakan dalam sepakbola [32] [33].

1. *Minutes Played* (Jumlah menit seorang pemain bermain)
2. *Goals* (Jumlah *goal* yang dicetak)
3. *Assist* (Jumlah *assist* yang dicetak)
4. *Goals Conceded* (Jumlah *goals conceded* saat pemain tersebut bermain)
5. *Own Goals* (Jumlah seorang pemain mencetak *goal* ke gawang sendiri)
6. *Yellow Card* (Kartu Kuning yang diterima)
7. *Red Card* (Kartu Merah yang diterima)
8. *Total Touches* (Total pemain menyentuh bola)
9. *Total Touches Final 3rd* (Total pemain menyentuh bola di lapangan bagian 1/3 musuh)
10. *Total Touches Opposition Half* (Total pemain menyentuh bola di lapangan bagian musuh)
11. *Min/Touches* (Jumlah menit bermain dibagi jumlah pemain menyentuh bola)
12. *Pass Received Total* (Total pemain menerima *pass*)
13. *Pass Received Opposition Half* (Total pemain menerima *pass* di

half musuh)

14. *Pass Received Final 3rd* (Total pemain menerima bola di lapangan bagian 1/3 musuh)
15. *Min/Pass Received* (Jumlah menit bermain dibagi jumlah pemain menerima bola)
16. *Take Ons* (Total pemain mencoba melewati pemain musuh menggunakan *dribble*)
17. *Successful Take Ons* (Total pemain sukses melewati pemain musuh menggunakan *dribble*)
18. *Successful Take Ons Percentage* (Total “Successful Take Ons” dibagi “Total Take Ons”)
19. *Total Pass* (Total pass yang dilakukan pemain)
20. *Total Successful Pass* (Total pass yang sukses dilakukan pemain)
21. *Successful Pass Percentage* (Total “Successful Pass” dibagi “Total Pass”)
22. *Chance Created* (Total pemain menghasilkan *chance* untuk pemain lain mencetak goal)
23. *Big Chance Created* (Total pemain menghasilkan *chance* untuk pemain lain mencetak goal, dimana *chance* tersebut cukup diharapkan untuk mencetak gol seperti skenario *one on one* atau tendangan dari jarak dekat)
24. *Min/Chance Created* (Jumlah menit bermain dibagi jumlah *chance*

- created*)
25. *Min/Goals* (Jumlah menit bermain dibagi dengan jumlah *goal*)
 26. *Shot Accuracy* (Jumlah *Attempts on Target* dibagi dengan jumlah *Attempts*)
 27. *Goal Conversion* (Jumlah *goal* dibagi dengan jumlah *total attempts*)
 28. *Total Attempts* (Total pemain menendang atau menyundul bola kearah gawang)
 29. *Total Attempts on Target* (Total pemain menendang atau menyundul bola kearah gawang dari dalam kotak penalty)
 30. *Min/Attempts* (Jumlah menit bermain dibagi dengan jumlah *attempts*)
 31. *Total Aerial Duels* (Total pemain mempertarungkan bola di udara dengan pemain musuh)
 32. *Total Successful Aerial Duels* (Total pemain sukses merebut bola di udara dari pemain musuh)
 33. *Successful Aerial Duels Percentage* (Total “Successful Aerial Duel” dibagi “Total Aerial Duel”)
 34. *Total Tackles* (Total pemain mencoba merebut bola dari musuh)
 35. *Total Successful Tackles* (Total pemain berhasil merebut bola dari musuh)
 36. *Successful Tackles Percentage* (Total “Successful Tackles” dibagi “Total Tackles”)

37. *Interceptions* (Total pemain mencegat *passing* musuh)
38. *Recoveries* (Situasi dimana seorang pemain mengambil bola untuk tim)
39. *Clearances* (Situasi dimana seorang pemain menendang bola jauh dari gawang sendiri)
40. *Blocks* (Total pemain menghadang tendangan musuh)
41. *Errors leading to Chance* (Total kesalahan pemain yang menghasilkan *chance* bagi musuh)
42. *Errors leading to Goals* (Total kesalahan pemain yang menghasilkan goal bagi musuh)

Agar data tersebut dapat dianalisa dalam konteks yang sama, beberapa data memiliki variasi seperti *Attempts/Minutes Played*, dimana jumlah *shot* dibagi jumlah menit pemain tersebut bermain.

Tabel 2.3. Contoh statistik sepakbola tim Sheffield United pada musim 20/21 (understat.com)

Nama	Posisi	Apps	Min	Goal	Assist	Shot/Min	KP/Min
David McGoldrick	F	9	480	2	0	1.88	0.38
Billy Sharp	F	4	124	1	0	2.18	0.00
Sander Berge	M	9	733	1	1	0.86	1.23
Oliver McBurnie	F	9	541	0	0	2.16	1.00
Oliver Burke	F	4	255	0	0	1.41	1.06
Rhian Brewster	F	5	260	0	0	0.69	1.05

2.26 Penelitian Terdahulu

Tabel 2.4. Penelitian Terdahulu 1.

Nama Penulis	Pires, Marcelo Santon, Vitor
Nama Jurnal, Vol, Edisi, Tahun	Journal of Sports Science, Edisi 6, 2018
Judul Artikel	Assessing the Impact of Internet of Everything Technologies in Football
Permasalahan	Mengetahui pengaruh dari <i>internet of things</i> dalam dunia sepakbola.
Metode	Studi Kasus, Kuisisioner
Hasil	Berbagai gadget seperti microchip yang merekam gerak pemain dalam pertandingan dapat memiliki peran penting di masa depan.
Kesimpulan	Permainan sepakbola memiliki banyak kesempatan untuk berkembang, terutama dalam penggunaan teknologi
Link	https://run.unl.pt/bitstream/10362/30073/1/TGI%200120.pdf

Tabel 2.5. Penelitian Terdahulu 2.

Nama Penulis	Blobel, Thomas Lames, Martin
Nama Jurnal, Vol, Edisi, Tahun	International Journal of Computer Science in Sport, Volume 19: Issue 1, 2020
Judul Artikel	A Concept for Club Information Systems (CIS) - An Example for Applied Sports Informatics
Permasalahan	Mengetahui konsep pengembangan sistem informasi menggunakan <i>model driven architecture</i>
Metode	Studi Kasus, Studi Literatur

Hasil	Terbuatnya sebuah <i>model driven architecture</i> yang dibuat berdasarkan <i>three-layer architecture</i> untuk sistem BI, yang disesuaikan dengan keperluan tim sepakbola.
Kesimpulan	Dibuatnya sebuah konsep CSI (Club Information System) yang dapat dijadikan acuan dalam pembuatan sistem informasi untuk tim sepakbola
Link	https://content.sciendo.com/view/journals/ijcss/19/1/article-p102.xml?language=en

Tabel 2.6. Penelitian Terdahulu 3.

Nama Penulis	Fitri, Indah Fadilla, Noufal Harsa, Awang
Nama Jurnal, Vol, Edisi, Tahun	Prosiding Seminar Nasional Ilmu Komputer dan Teknologi Informasi. Desember 2018
Judul Artikel	Sistem Pendukung Keputusan Penempatan Posisi Pemain Sepak Bola Dengan Penerapan Metode Simple Additive Weighting
Permasalahan	Menganalisa mengenai penggunaan metode SAW dalam mencari posisi terbaik pemain sepakbola
Metode	Studi Kasus
Hasil	Mengetahui cara kerja dan proses implementasi metode SAW dalam mencari posisi terbaik pemain sepakbola
Kesimpulan	SAW dapat digunakan untuk mengetahui posisi terbaik pemain sepakbola.
Link	https://garuda.ristekbrin.go.id/documents/detail/906396

Tabel 2.7. Penelitian Terdahulu 4.

Nama Penulis	Nurzahputra, Aldi Rizqi Pranata, Afrizal Puwinarko, Aji
Nama Jurnal, Vol, Edisi, Tahun	SISTEMASI 8(1) (2019)
Judul Artikel	Sistem Pendukung Keputusan Penentuan Starting Line Up Tim Sepakbola Menggunakan Metode Perbandingan Eksponensial (MPE)
Permasalahan	Membuat sistem dengan metode SAW untuk membantu pemilihan pemain bola dalam suatu pertandingan
Metode	SAW
Hasil	Terdapat hasil metode SAW dimana 4 pemain dibandingkan berdasarkan kategori dan bobot yang telah ditetapkan. Hasil dari kalkulasi tersebut menghasilkan sebuah nilai yang dapat digunakan untuk membandingkan performa pemain.
Kesimpulan	Metode SAW dapat digunakan untuk menentukan line up sebuah tim sepakbola.
Link	https://media.neliti.com/media/publications/90202-ID-artikel-dari-jtsiskom-test.pdf

Tabel 2.8. Penelitian Terdahulu 5

Nama Penulis	Setiawan, Johan Raden Ajeng Yosua Ariane Amos Wiseso
Nama Jurnal, Vol, Edisi, Tahun	VOL 9 NO 1 (2018): ULTIMA INFOSYS : JURNAL ILMU SISTEM INFORMASI
Judul Artikel	Sistem Pendukung Keputusan Dalam Penentuan Penerima Beasiswa PT BFI Finance Indonesia Tbk Menggunakan Metode Profile Matching
Permasalahan	Proses pembuatan sistem pendukung keputusan dalam penentuan penerima beasiswa PT BFI Finance Indonesia menggunakan metode Profile Matching
Metode	Profile Matching
Hasil	Sistem yang terbuat menghasilkan tingkat akurasi sebesar 98.4% dari 500 kandidat.
Kesimpulan	Profile Matching dapat digunakan sebagai salah satu metode dalam sistem pengambilan keputusan.
Link	https://ejournals.umn.ac.id/index.php/SI/article/view/856

Berdasarkan penelitian terdahulu pada tabel 2.3, 2.4, 2.5, 2.6 dan 2.7 dapat disimpulkan bahwa berbagai materi yang dijelaskan dalam jurnal-jurnal tersebut dapat digunakan dalam membangun sistem untuk tim sepakbola. Keunggulan dari sistem yang dibuat adalah menggabungkan berbagai fitur dalam satu sistem informasi. Keputusan tersebut dibuat dengan tujuan untuk memudahkan pelatih agar dapat menganalisa banyak hal dalam suatu *platform* yang sama. Penelitian terdahulu berjudul “A Concept for Club Information Systems (CIS) - An Example for Applied Sports Informatics” akan digunakan sebagai acuan *user requirement*

dan struktur dalam pembuatan sistem. Untuk meningkatkan *versatility* sistem, ditambahkan juga sebuah sistem pendukung keputusan menggunakan metode SAW untuk mengukur posisi terbaik seorang pemain berdasarkan data kualitatif dari pelatih.