

BAB 4

HASIL PENELITIAN

4.1 RINGKASAN HASIL PENELITIAN

Ringkasan hasil penelitian ini menunjukkan bahwa SPK yang dikembangkan menggunakan metode SAW berhasil mengoptimalkan proses pemilihan pemain untuk tim MLBB. Berdasarkan pengujian, SPK mampu mempertimbangkan berbagai faktor penting seperti kemampuan individu, komunikasi tim, dan kekompakan, yang sebelumnya sering diabaikan dalam proses seleksi manual. Penggunaan SPK ini meningkatkan dalam memilih pemain dengan kemampuan optimal untuk *role jungler*. Sistem ini juga memberikan panduan terarah untuk pembentukan tim yang lebih solid dan strategi yang lebih baik.

4.2 ANALISIS SISTEM

4.2.1 Kebutuhan Fungsional

Persyaratan sistem sangat dibutuhkan untuk memastikan bahwa semua komponen yang diperlukan tersedia dan berfungsi sesuai dengan yang diharapkan. Analisis ini membantu dalam merencanakan implementasi sistem serta memastikan bahwa tujuan proyek dapat tercapai dengan baik. Analisis kebutuhan untuk sistem yang dikembangkan:

1. Sistem mampu mengolah data pemain.
2. Sistem mampu melakukan perhitungan terkait pemain.
3. Sistem memiliki fungsi CRUD (*Create, Read, Update, Delete*).
4. Sistem dapat menampilkan hasil perhitungan.

4.2.2 Analisis Data

Data pemain yang dilengkapi kriteria yang menjadi acuan. Tabel 4.1 menyajikan data yang dikumpulkan pada *season* ke-31.

Tabel 4.1 Data Pemain

No	Nama Pemain	Avg Gpm	Avg Kda	Avg Damage	Avg Damage Diterima	Komunikasi	Team Fight	Turn Yg Dijuarai	Hero Power	Nama Hero	Pertandingan	Win Rate
1	Fakhruddin Abdul Aziz	666	3.67	3210	16646	80	50.9%	3	2431	Gusion	169	63.9%
2	Muhammad Andre Wijaya	618	3.03	3394	14148	80	50.5%	2	1357	Roger	190	55.3%
3	Angga Yudha Mulyono	509	3.08	2384	17106	85	50.7%	5	4753	Natalia	880	58.5%
4	Muhammad Rayhan	609	2.46	2904	21303	80	46.6%	0	2412	Martis	320	54.1%
5	Miqdam Syadda Lisidiq	525	3.39	2597	16716	85	50.9%	0	2348	Hayabusa	205	57.1%
6	Hadi Kostaman	560	3.82	2612	14083	80	52.7%	1	2350	Akai	258	69.4%
7	Maulana Ahmad Rivai	578	3.10	3107	19497	90	49.7%	0	4087	Julian	351	64.1%
8	Ilham Nur Fikri	663	3.39	2975	14173	90	37.9%	0	1075	Thamuz	102	64.1%

9	Jiva Saucha	628	2.83	3537	15198	80	52.7%	0	1893	Gusion	806	57.1%
10	Galang	573	3.19	2714	16359	85	54.5%	0	1372	Alucard	116	64.7%
11	Reza	627	3.18	3305	16379	80	50.9%	0	1270	Alpha	110	56.5%
12	Arif Rahman	570	3.12	3153	12241	80	49.7%	0	1297	Karrie	176	76.1%

Dari Tabel 4.1, terdapat 12 data yang akan digunakan untuk menentukan keputusan pemain terbaik dalam *role jungler*. Langkah pertama dalam proses perhitungan SAW adalah menetapkan kriteria dan bobot yang akan dievaluasi. Tahap berikutnya yaitu menentukan alternatif yang akan dievaluasi terhadap kriteria yang telah ditetapkan., pembuatan matriks keputusan dan normalisasi, serta langkah terakhir dalam proses ini adalah melakukan perangkingan untuk menentukan nilai terbesar. Contoh tahapan perhitungan dapat dilihat dalam urutan sebagai berikut:

1. Menentukan Kriteria dan Bobot

Menetapkan kriteria dan bobot yang akan digunakan sebagai acuan dalam pengambilan keputusan (C_i) serta alternatif yang akan dievaluasi (A_i).

- **Kriteria:**

- *Average Gold Per Menit* (C_1)
- *Average Kill Death Assist* (C_2)
- *Average Damage* (C_3)
- *Average Damage Diterima* (C_4)
- Komunikasi (C_5)
- *Team Fight* (C_6)
- Turnamen Yang Dijuarai (C_7)
- *Hero Power* (C_8)
- Pertandingan (C_9)
- *Win Rate* (C_{10})

Sebelum melakukan perhitungan tentukan terlebih yang akan menjadi kriteria *cost* atau *benefit*. Kriteria **benefit** pada $C_1, C_2, C_3, C_5, C_6, C_7, C_8, C_9, C_{10}$, sedangkan kriteria **cost** C_4 .

2. Menetapkan nilai dan bobot untuk setiap kriteria

Pada tahap ini, nilai untuk setiap kriteria diisi berdasarkan data yang telah dikumpulkan. Pembobotan diberikan menggunakan skala nilai dari 0 hingga 1.

Tabel 4.2 Nilai Alternatif di Setiap Kriteria

Alternatif	Kriteria									
	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
A1	3	2	2	2	4	1	4	3	1	3

- Pembobotan (w)

Pembobotan ini merujuk pada penentuan bobot untuk setiap kriteria, penentuan bobot untuk memilih pemain terbaik dalam MLBB adalah sebagai berikut:

Tabel 4.3 Pembobotan Kriteria

Kriteria	Bobot
C1	0,25
C2	0,15
C3	0,1
C4	0,02
C5	0,04
C6	0,1
C7	0,04
C8	0,1
C9	0,1
C10	0,1
Total	1

- Alternatif

- Fakhrudin Abdul Aziz (A1)

3. Normalisasi Metrik (r)

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\max x_{ij}} & \text{jika } j \text{ atribut menguntungkan (benefit)} \\ \frac{\min x_{ij}}{x_{ij}} & \text{jika } j \text{ atribut biaya (cost)} \end{cases} \quad (4)$$

- **Kriteria C1**

$$r_{11} = \frac{3}{\max\{3;2;2;2;2;2;2;3;2;2;2\}} = \frac{3}{3} = 1$$

- **Kriteria C2**

$$r_{12} = \frac{2}{\max\{2;2;2;1;2;2;2;1;2;2\}} = \frac{2}{2} = 1$$

- **Kriteria C3**

$$r_{13} = \frac{2}{\max\{2;2;1;1;1;1;2;1;2;2\}} = \frac{2}{2} = 1$$

- **Kriteria C4**

$$r_{14} = \frac{\min\{2;3;2;1;2;3;2;3;3;2;2;3\}}{2} = \frac{1}{2} = 0,5$$

- **Kriteria C5**

$$r_{15} = \frac{4}{\max\{4;4;4;4;4;4;4;4;4;4\}} = \frac{4}{4} = 1$$

- **Kriteria C6**

$$r_{16} = \frac{1}{\max\{1;1;1;1;1;2;1;1;2;2;1;1\}} = \frac{1}{2} = 0,5$$

- **Kriteria C7**

$$r_{17} = \frac{4}{\max\{4;3;5;1;1;2;1;1;1;1;1\}} = \frac{4}{5} = 0,8$$

- **Kriteria C8**

$$r_{18} = \frac{3}{\max\{3;2;5;3;3;3;5;2;2;2;2\}} = \frac{3}{5} = 0,6$$

- **Kriteria C9**

$$r_{19} = \frac{1}{\max\{1;1;2;1;1;1;1;1;2;1;1\}} = \frac{1}{3} = 1$$

- **Kriteria C10**

$$r_{110} = \frac{3}{\max\{3;2;2;2;2;3;3;3;2;3;2;4\}} = \frac{3}{4} = 0,75$$

Hasil Normalisasi

$$R = [1 \quad 1 \quad 1 \quad 0,5 \quad 0,8 \quad 0,6 \quad 0,5 \quad 0,75 \quad]$$

4. Proses Perangkingan

- Proses perangkingan dengan mengalikan nilai sesuai dengan bobot yang telah ditetapkan

$$w : [0,25 \quad 0,15 \quad 0,1 \quad 0,02 \quad 0,04 \quad 0,1 \quad 0,04 \quad 0,1 \quad 0,1 \quad 0,1]$$

- **Berikut adalah hasil yang diperoleh:**

$$\begin{aligned}
 V_1 &= (0,25) (1) + (0,15) (1) + (0,1) (1) + (0,02) (0,5) + (0,04) (1) + (0,1) \\
 &\quad (0,5) + (0,04) (0,8) + (0,1) (0,6) + (0,1) (0,5) + (0,1) (0,75) \\
 &= 0,25 + 0,15 + 0,1 + 0,01 + 0,04 + 0,05 + 0,032 + 0,06 + 0,05 + 0,075 \\
 &= 0,817
 \end{aligned}$$

5. Melakukan Perangkingan

Melakukan perangkingan dari semua alternatif.

6. Menentukan Alternatif Terbaik

Menentukan alternatif terbaik berdasarkan alternatif yang memiliki nilai terbesar.

4.3 PERANCANGAN SISTEM

4.3.1 Perancangan Use Case

Use Case diagram dirancang untuk mengidentifikasi aktor-aktor yang terlibat dalam sistem yang akan dibuat. Dalam sistem yang akan dikembangkan, hanya akan ada satu aktor, yaitu admin. *Use case* SPK disajikan pada bagian lampiran. Tabel 4.4 akan menguraikan definisi dari setiap *use case* yang ada.

Tabel 4.4 Use Case SPK

No	Use Case	Keterangan
1	Mengelola Pemain	Prosedur yang diterapkan untuk mengurus informasi para pemain.
2	Input Data Pemain	Proses ini merujuk pada tindakan untuk memasukkan data para pemain ke dalam sistem.
3	Melihat Hasil	Langkah ini mencakup tindakan untuk meninjau prestasi para pemain yang paling unggul.

4.3.2 Perancangan Activity Diagram

Activity Diagram (Diagram aktivitas) adalah suatu desain yang bertujuan untuk memvisualisasikan rangkaian aktivitas dari sebuah proses yang terdapat

dalam diagram *use case*. Setiap langkah dalam proses tersebut akan diwakili dengan simbol awal (*start*) hingga simbol akhir dari aktivitas yang bersangkutan.

Proses dimulai saat admin memilih opsi "Tambah Data" dari menu. Sistem akan menampilkan formulir untuk memasukkan informasi pemain. Admin akan mengisi data tersebut, lalu sistem akan melakukan validasi. Apabila validasi sukses, data akan disimpan. Namun, jika validasi gagal, proses akan kembali ke tahap pengisian data oleh admin. Gambar *activity diagram* proses mengelola pemain disajikan di bagian lampiran.

4.3.3 Perancangan Class Diagram

Class diagram adalah suatu desain yang bertujuan untuk mengilustrasikan keterkaitan antara berbagai objek yang ada dalam sistem yang sedang dikembangkan. Fungsinya adalah untuk menyederhanakan proses implementasi dengan mengelompokkan berbagai jenis objek yang ada. Setelah itu, objek-objek tersebut didefinisikan dengan atribut-atribut dan metode-metode yang dapat dilakukan.

Perancangan *class diagram* pada sistem ini hanya mempunyai satu kelas yaitu kelas kriteria. Perancangan *class diagram* yang menunjukkan struktur objek-objek dalam sistem yang sedang dikembangkan disajikan di bagian lampiran. Keterangan *class diagram* dapat dilihat pada Tabel 4.5.

Tabel 4.5 Keterangan Class Diagram

No	Nama Class	Keterangan
1	Kriteria	Kelas ini digunakan untuk melakukan pengolahan dasar pada data, seperti menambah data (<i>add</i>), membaca (<i>read</i>), memperbarui (<i>update</i>), dan menghapus (<i>delete</i>).

4.4 PERANCANGAN BASIS DATA

Pada perancangan skema ini, setiap dokumen dalam *collection* kriteria pemain dan menyimpan atribut-atribut penting seperti id, nama pemain, *average gold* per menit, *average KDA* (*Kill Death Assist*), *average damage*, *average damage* diterima, *team fight*, turnamen yang dijuarai, *hero power*, nama *hero*,

pertandingan, *win rate*, dan komunikasi. Skema dokumen MongoDB yang dirancang disajikan pada bagian lampiran.

4.5 PERANCANGAN ANTARMUKA

Dalam perancangan antarmuka, berbagai desain *visual* dan tata letak antarmuka pengguna yang telah dirancang akan disajikan pada bagian lampiran.

4.6 IMPLEMENTASI SISTEM

Pada tahap ini, penting untuk memastikan bahwa basis data dan program telah sesuai dengan rencana yang telah dianalisis pada bab sebelumnya. Tujuan dari tahap implementasi adalah untuk menyiapkan dan melaksanakan semua kegiatan yang diperlukan agar sistem dapat diterapkan sesuai dengan desain yang telah ditetapkan sebelumnya. Setelah itu, dilakukan tahapan pengujian sistem untuk meminimalkan kemungkinan kesalahan yang mungkin terjadi.

4.7 PEMBAHASAN BASIS DATA

Pembuatan basis data dan *collection* adalah langkah awal yang penting dalam implementasi suatu sistem. Basis data dan *collection* tersebut akan menjadi dasar yang digunakan untuk menyimpan dan mengelola data yang dibutuhkan oleh sistem.

4.7.1 Pembahasan Collection

Setelahnya, *collection* yang diperlukan untuk sistem akan dibentuk. Berdasarkan rencana yang telah ditetapkan, terdapat dua *collection* yang akan dijalankan.

4.7.2 Collection Kriteria Pemain

Collection kriteria pemain terdiri dari 13 *field* yaitu id kriteria admin, nama pemain, *average gold* per menit, *average KDA* (*Kill Death Assist*), *average damage*, *average damage* diterima, *team fight*, turnamen yang dijuarai, komunikasi, nama *hero*, *hero power*, pertandingan, dan *win rate*. Gambar *collection* kriteria pemain disajikan pada lampiran.

4.8 PEMBUATAN SISTEM

Sebuah sistem telah dibuat dengan memanfaatkan bahasa pemrograman *Python* dan basis data MongoDB. Setelah berhasil *login*, admin memiliki

kemampuan untuk mengelola data di dalam sistem ini dengan melakukan perubahan, penambahan, dan penghapusan sesuai kebutuhan.

4.8.1 Koneksi Basis Data

Sebelum mengakses data dalam basis data, langkah pertama yang diperlukan adalah melakukan encode pada basis data tersebut. Dalam aplikasi ini, disarankan untuk membuat sebuah kode program koneksi yang dapat diakses dari berbagai halaman. Berikut kode program yang dapat digunakan untuk menghubungkan ke basis data.

```
URL_ENDPOINT = os.getenv('URL_ENDPOINT')
API_KEY = os.getenv('API_KEY')
DATA_SOURCE = os.getenv('DATA_SOURCE')
DATABASE = os.getenv('DATABASE')
COLLECTION = os.getenv('COLLECTION')
```

4.8.2 Setting Modul Server

Modul server digunakan untuk memuat semua fungsi atau *library* yang dibutuhkan dalam sistem ini. Ini mencakup *library* yang sudah digunakan serta halaman-halaman yang terlibat dalam pembuatan sistem.

```
certifi==2022.12.7
charset-normalizer==3.1.0
click==8.1.3
colorama==0.4.6
dnspython==2.3.0
Flask==2.2.3
idna==3.4
itsdangerous==2.1.2
Jinja2==3.1.2
MarkupSafe==2.1.2
pymongo==4.3.3
python-dotenv==1.0.0
requests==2.28.2
urllib3==1.26.15
Werkzeug==2.2.3
```

4.8.3 Kode Program Modul Pemain

Kode program modul pemain ini bertugas untuk mengambil atau memproses data dalam sistem. Fungsi utama dari modul ini adalah menghubungkan basis data untuk menjalankan perintah seperti menyimpan, mengubah, menghapus, dan menampilkan data.

```
def getAllDataTraining():
    if 'username' and 'password' in session:
        data = {}
        _id = []
        url = URL_ENDPOINT+"/action/find"
        payload = json.dumps({
            "dataSource": DATA_SOURCE,
            "collection": COLLECTION,
            "database": DATABASE,
            "filter": {}
        })
        response = requests.request("POST",
url,headers=headers, data=payload)
        json_response = response.json()
        if json_response != 'undefined':
            documents = json_response['documents']
            # print(documents)
            for doc in documents:
                data[doc['_id']]={
                    'C1':float(doc['avg_gpm']),
                    'C2':float(doc['avg_kda']),
                    'C3':float(doc['avg_damage']),
                    'C4':float(doc['avg_dmg_diterima']),
                    'C5':float(doc['komunikasi']),
                    'C6':float(doc['team_fight']),
                    'C7':float(doc['tourn_yg_dijuarai']),
                    'C8':float(doc['hero_power']),
                    'C9':float(doc['pertandingan']),
                    'C10':float(doc['win_rate']),
                }
                _id.append({doc['_id'] : doc['_id']})
            return {'data':data, 'id':_id}
        return redirect('/login')
```

4.8.4 Kode Program Routing Data Pemain

Kode program *routing* dalam data pemain bertugas untuk membuat URL yang diperlukan. Proses pembuatan *routing* ini mengikuti mekanisme penamaan berdasarkan fungsinya, sehingga memudahkan admin dalam mengakses sistem.

```

@app.route('/', methods=['GET', 'POST'])
def home():
    if 'username' and 'password' in session:
        return render_template('home.html')
    return redirect('/login')
@app.route("/tambah-data", methods=['POST'])
def tambah_data():
@app.route("/edit-data", methods=['POST'])
def edit_data():
@app.route('/get-one-data/<string:id>', methods=['POST'])
def get_one_data(id):
@app.route('/login', methods=['GET', 'POST'])
def login():
@app.route('/logout')
def logout():
@app.route('/indexData', methods=['POST'])
def index_data():
@app.route('/delete-one/<string:id>', methods=['DELETE'])
def delete_one(id):
@app.route("/metodesaw", methods=['GET', 'POST'])
def metode_saw():
@app.route("/indexRanking", methods=['POST'])
def index_ranking():

```

4.8.5 Kode Program Perhitungan SAW

Kode program perhitungan SAW ini berfungsi untuk mengevaluasi alternatif berdasarkan kriteria-kriteria tertentu dengan mengikuti proses normalisasi data dan perhitungan bobot. Pertama, data alternatif dinormalisasi agar memiliki skala yang sama, kemudian bobot untuk setiap alternatif dihitung dengan mengalikan nilai normalisasi dengan bobot kriteria yang telah ditetapkan sebelumnya. Hasil dari kode program ini adalah nilai akhir SAW untuk setiap alternatif, yang dapat digunakan dalam pengambilan keputusan.

```

@app.route("/indexRanking", methods=['POST'])
def index_ranking():
    for i in paramKriteria:
        test['C' + str(j)] = {'rating': i['rating'],
        'atribut': i['atribut']}
        j += 1

```

```

status = False
res = []
allData = getAllDataTraining()
data = allData["data"]
if len(data) != 0:
    status = True
    kriteria = test
    # mendapatkan rating kriteria
    rating = [kriteria[i]['rating'] for i in
kriteria.keys()]
    # normalisasi bobot
    bobot = [val/sum(rating) for val in rating]
    C_MaxMin = []
    #perulangan utk dpt atribut kriteria
    for key in kriteria.keys():
        C = [
            data[i][key] for i in data.keys()
        ]
        #periksa atribut apakah cost / benefit
        if kriteria[key]['atribut'] == 'benefit':
            C_MaxMin.append(max(C))
        else:
            C_MaxMin.append(min(C))
    #normalisasi
    norms = []
    v = []
    n=0
    for vals in data.values():
        norm = []
        i = 0
        vn = 0
        #langkah mendapat atribut kriteria
        for key, val in vals.items():
            #cek atribut apakah cost / benefit
            if kriteria[key]['atribut'] ==
'benefit':
                n = val/C_MaxMin[i]
#menyimpan normalisasi dlam var n
            else:
                C_MaxMin.append(min(C))
        #menghitung vn
        vn += (n * bobot[i])
        norm.append(n)
        i+=1
    #simpan hasil normalisasi

```

```

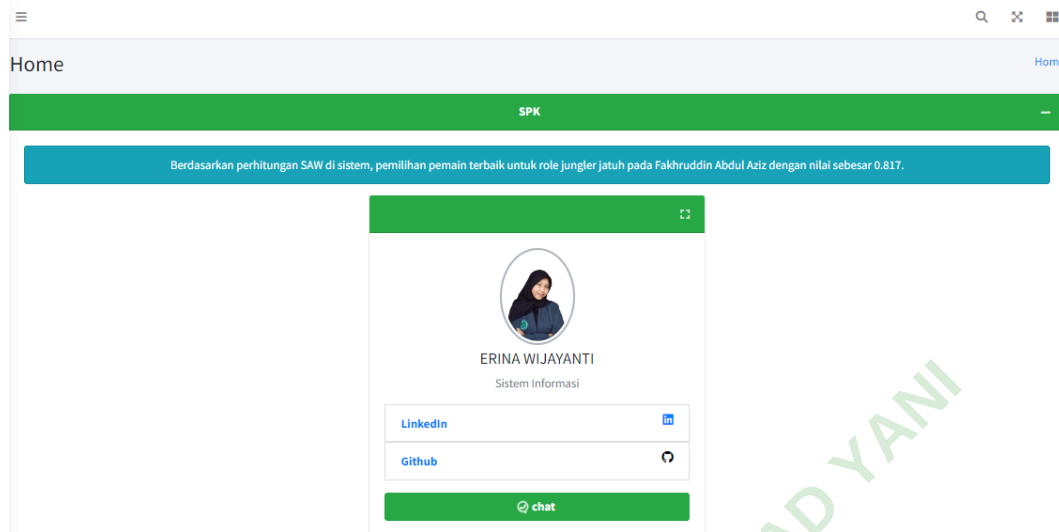
        norms.append(norm)
        #simpan vn ke dalam vektor v
        v.append(round(vn,3))
    #langkah perangkingan
    rank = {}
    i = 0
    for key in data.keys():
        rank[key] = v[i]
        i+=1
    sorted_rank = sorted(
        [
            (value, key) for (key, value) in
rank.items()
        ], reverse=True
    )
    for i in sorted_rank:
        cek = getData(i[1])
        res.append([i[0],cek])
    return jsonify({'status': status,'res': res,})
return redirect('/login')

```

4.9 PEMBAHASAN ANTARMUKA PROGRAM

4.9.1 Halaman Utama

Setelah berhasil mengakses, admin akan dialihkan ke halaman utama sistem. Pada halaman ini, terdapat berbagai menu dan fitur, termasuk menu metode yang mencakup metode SAW. Dalam halaman utama ini pemain terbaik dengan nilai tertinggi ditampilkan. Gambar 4.1 menunjukkan tampilan halaman utama.



Gambar 4.1 Halaman Utama

4.9.2 Halaman SAW

Halaman menu SAW adalah halaman yang digunakan untuk perhitungan, di mana admin dapat memasukkan, mengedit, dan menghapus data. Untuk melihat hasil perbandingan, admin hanya perlu menggulir ke bawah. Gambar 4.2 menunjukkan tampilan halaman SAW.

Nama Pemain	Average GPM	Average KDA	Average Damage	Average Damage Diterima	Komunikasi	Team Fight(%)	Turn Yang Dijuarai	Nama Hero	Hero Power	Pertandingan	Win Rate(%)	Aksi
Angga Yudha Mulyono	509	3.08	2384	17106	80	50.7	5	Natalia	4753	880	58.5	[Edit][Delete]
Arif Rahman	570	3.12	3153	12241	80	49.7	0	Karie	1297	176	76.1	[Edit][Delete]
Fakhruddin Abdul Aziz	666	3.67	3210	16646	80	50.9	3	Gusion	2431	169	63.9	[Edit][Delete]

Gambar 4.2 Halaman Metode SAW

4.9.3 Halaman Tambah Data Pemain

Admin menggunakan halaman tambah data untuk memasukkan data baru ke dalam sistem. Di sini, admin dapat mengisi informasi yang diperlukan dan menyimpannya. Gambar 4.3 menunjukkan tampilan halaman tambah data.

Tambah Data	
Nama Pemain Wajib diisi	Average GPM Wajib diisi
Ex : 600	
Average KDA Wajib diisi	Average Damage Wajib diisi
Ex : 3.25	Ex : 3210
Average Damage Diterima Wajib diisi	Komunikasi Wajib diisi
Ex : 15600	Ex : 80
Team Fight Wajib diisi	Turn Yang Dijuarai Wajib diisi
Ex : 50.9	Ex : 1
Nama Hero Wajib diisi	Hero Power Wajib diisi
Ex : Lancelot	Ex : 3345
Pertandingan Wajib diisi	Win Rate Wajib diisi
Ex : 120	Ex : 65.3

+
add data

Gambar 4.3 Tambah Data

4.9.4 Halaman Edit Pemain

Pada halaman ini, admin dapat melakukan pengeditan data yang telah ada di dalam sistem. Admin dapat memperbarui data atau mengoreksi kesalahan. Semua perubahan yang dilakukan harus disimpan agar data terbaru tercatat dalam basis data. Gambar 4.4 memperlihatkan halaman tampilan *edit* pemain.

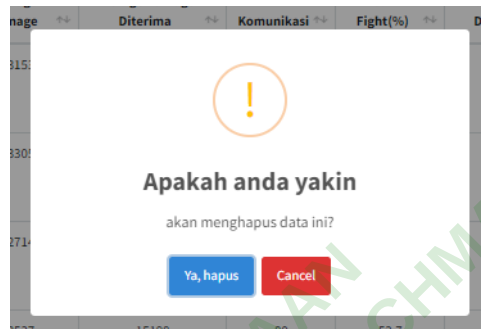
Edit Data	
Nama Pemain	Avg GPM
Arif Rahman	570
	Ex : 600
Avg KDA	Avg Damage
3.12	3153
Ex : 3.25	Ex : 3210
Avg Damage Diterima	Komunikasi
12241	80
Ex : 15600	Ex : 80
Team Fight	Turn Yg Dijuarai
49	0
Ex : 50.9	Ex : 1
Nama Hero	Hero Power
Karrie	1297
Ex : Lancelot	Ex : 3345
Pertandingan	Win Rate
176	76
Ex : 120	Ex : 65.3

Close Edit Data

Gambar 4.4 Edit Data

4.9.5 Halaman Hapus Data

Halaman hapus data dirancang untuk memungkinkan pengguna menghapus data pemain dari basis data melalui antarmuka web. Halaman ini menampilkan daftar pemain yang ada dalam basis data, masing-masing dengan opsi untuk dihapus. Gambar 4.5 menunjukkan tampilan halaman tambah data.



Gambar 4.5 Hapus Data

4.9.6 Halaman Perangkingan

Halaman perangkingan adalah halaman yang menampilkan hasil akhir dari proses perangkingan berdasarkan data yang telah diinput dan dihitung. Admin dapat melihat urutan peringkat langsung pada halaman ini. Tampilan yang menampilkan hasil perangkingan dapat ditemukan dalam Gambar 4.6.

Ranking ↑↓	Nilai perhitungan ↑↓	Nama Pemain ↑↓	Average Damage ↑↓	Average KDA ↑↓	Average Damage Diterima ↑↓	Komunikasi ↑↓	Team Fight(%) ↑↓	Turn Yg Dijuarai ↑↓	Nama Hero ↑↓	Hero Power ↑↓	Pertandingan ↑↓	Win Rate(%) ↑↓
1	0.817	Fakhruddin Abdul Aziz	3210	3.67	16646	80	50.9	3	Gusion	2431	169	63.9
2	0.757	Angga Yudha Mulyono	2384	3.08	17106	85	50.7	5	Natalia	4753	880	58.5
3	0.75	Maulana Ahmad Rivai	3107	3.1	19497	90	49	0	Julian	4087	351	64.5
4	0.72	Ilham Nur Fikri	2975	3.39	14173	90	37.9	0	Thamuz	1075	102	64.5
5	0.714	Hadi Kostaman	2612	3.82	14083	80	52.7	1	Akai	2350	258	69.4
6	0.711	Arif Rahman	3153	3.12	12241	80	49	0	Karrie	1297	176	78.4
7	0.69	Galang	2714	3.19	16359	85	54.5	0	Alucard	1372	116	64.7

Gambar 4.6 Perangkingan

Alternatif	NORMALISASI										Perangkingan		
	Kriteria												
	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	v	Nilai	Rangking
A1	1	1	1	0,5	1	0,5	0,8	0,6	0,5	0,75	v1	0,817	1
A2	0,666666667	1	1	0,333333333	1	0,5	0,6	0,4	0,5	0,5	v2	0,677	9
A3	0,666666667	1	0,5	0,5	1	0,5	1	1	1	0,5	v3	0,787	2
A4	0,666666667	0,5	0,5	1	1	0,5	0,2	0,6	0,5	0,5	v4	0,570	12
A5	0,666666667	1	0,5	0,5	1	0,5	0,2	0,6	0,5	0,5	v5	0,635	11
A6	0,666666667	1	0,5	0,333333333	1	1	0,4	0,6	0,5	0,75	v6	0,714	5
A7	0,666666667	1	1	0,5	1	0,5	0,2	1	0,5	0,75	v7	0,750	3
A8	1	1	0,5	0,333333333	1	0,5	0,2	0,4	0,5	0,75	v8	0,720	4
A9	0,666666667	0,5	1	0,333333333	1	1	0,2	0,4	1	0,5	v9	0,686	8
A10	0,666666667	1	0,5	0,5	1	1	0,2	0,4	0,5	0,75	v10	0,690	7
A11	0,666666667	1	1	0,5	1	0,5	0,2	0,4	0,5	0,5	v11	0,665	10
A12	0,666666667	1	1	0,333333333	1	0,5	0,2	0,4	0,5	1	v12	0,711	6

Gambar 4.7 Hasil Perhitungan Excel

Berdasarkan hasil perhitungan di excel dan sistem, menunjukkan kesesuaian 100%.

4.10 TESTING SISTEM

Digunakan metode *Black Box Testing* pada pengujian sistem dengan pendekatan *Functional Testing*. Teknik ini melibatkan pengujian spesifikasi sistem melalui eksekusi fungsi-fungsi perangkat lunak. Hasil pengujian ini dievaluasi untuk memverifikasi kesesuaiannya dengan proses bisnis yang diinginkan. Jika hasil pengujian memenuhi standar, sistem akan beroperasi sesuai dengan yang diharapkan. Namun, jika terjadi kesalahan atau *error*, pesan *error* akan ditampilkan pada tampilan *interface framework*.

4.10.1 Hasil Pengujian Fungsi Manajemen Data Pemain

Tabel 4.6 Hasil Pengujian Fungsi Data Pemain

Function	Scenario	Hasil Yang Diharapkan	Keterangan
Data pemain	Klik menu SAW	Menampilkan informasi tentang para pemain.	Berhasil
Tambah data pemain	Klik tambah data	Penambahan data pemain berhasil dilakukan.	Berhasil
Ubah Data Pemain	Klik tombol <i>edit</i> data/ubah data	Perubahan data berhasil dilakukan sesuai dengan yang diinginkan.	Berhasil

Hapus Data Pemain	Klik tombol hapus	Data berhasil dihapus dari sistem.	Berhasil
Cari Data Pemain	Melakukan pencarian di kolom pencarian	Menampilkan data yang sesuai dengan kata kunci dan kriteria yang diminta.	Berhasil

4.10.2 Hasil Perangkingan

Tabel 4.7 Hasil Perangkingan

Function	Scenario	Hasil Yang Diharapkan	Keterangan
Hasil Perangkingan	Scroll ke bawah	Menampilkan hasil pemain	Berhasil
Mengurutkan berdasarkan abjad	Klik tombol panah atas bawah	Berhasil menampilkan data yang urut	Berhasil

Proses pengambilan keputusan yang belum terkomputerisasi membutuhkan lebih banyak waktu selama 2 hari dan harus membandingkan kriteria pemain yang satu dengan yang lain secara manual, yang melibatkan evaluasi mendetail dari berbagai aspek seperti performa individu dan kemampuan komunikasi. Setiap data harus dianalisis dan dibobotkan secara hati-hati untuk memastikan pemilihan pemain yang paling sesuai dengan kebutuhan tim. Cara ini tidak hanya memakan banyak waktu, tapi juga bisa membuat keputusan menjadi tidak objektif dan rentan terhadap kesalahan manusia..

Penggunaan metode SPK yang terkomputerisasi seperti SAW diharapkan dapat mempercepat proses dalam pengambilan keputusan. Implementasi sistem pendukung keputusan ini dapat mengurangi waktu yang dibutuhkan dan memberikan hasil yang lebih konsisten. Pada penentuan kriteria dilakukan dengan cara pengambilan data statistik performa pemain di Mobile Legend, dan untuk komunikasi dinilai oleh pelatih.

Setelah itu memberikan bobot pada setiap kriteria berdasarkan tingkat kepentingannya dalam konteks *role* dan tujuan tim. Kemudian melakukan perhitungan nilai akhir menggunakan metode SAW untuk menghasilkan peringkat pemain. Pada perhitungan menggunakan sistem hasil yang didapatkan lebih cepat, kurang dari 1 menit.

Pada proses pengerjaan sistem ini membutuhkan waktu 1 bulan lebih 14 hari. Peneliti mengalami beberapa kendala seperti kode program *error*, perhitungan manual dengan sistem berbeda dan kesulitan *deploy* di vercel. Akan tetapi setelah kesulitan itu *deploy* berhasil dijalankan

PERPUSTAKAAN
UNIVERSITAS JENDERAL ACHMAD YANI
YOGYAKARTA