

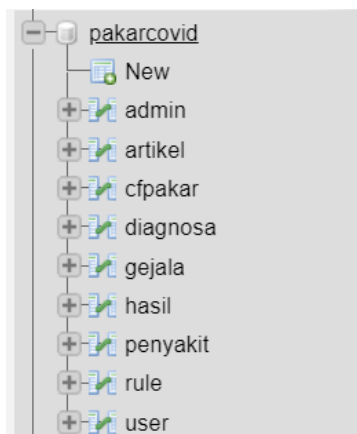
BAB V. IMPLEMENTASI DAN PENGUJIAN

5.1 Implementasi

Setelah dilakukan perancangan sistem, maka selanjutnya adalah implementasi sistem sesuai dengan perancangan yang dilakukan. Pada bagian ini menjelaskan tentang hasil dari sistem yang telah dibangun. Implementasi dijelaskan secara detail secara visual dengan tampilan gambar dan potongan kode program atau *listing code*, sebagai berikut :

5.1.1 Implementasi *Database*

Implementasi *database* sesuai dengan perancangan menggunakan *database MySQL* yang digunakan untuk menyimpan data dalam sistem, sebagai berikut :



Gambar 5.1.1 Implementasi Tabel *Database*

Sembilan tabel yang diimplementasikan dalam sistem tersebut yaitu, tabel admin, artikel, cfpakar, diagnosa, gejala, hasil, penyakit, rule dan user.

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> admin		1	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/> artikel		1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> cfpakar		6	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> diagnosa		57	InnoDB	utf8mb4_general_ci	48.0 KiB	-
<input type="checkbox"/> gejala		19	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> hasil		3	InnoDB	utf8mb4_general_ci	64.0 KiB	-
<input type="checkbox"/> penyakit		2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> rule		17	InnoDB	utf8mb4_general_ci	80.0 KiB	-
<input type="checkbox"/> user		8	InnoDB	utf8mb4_general_ci	16.0 KiB	-
9 tables	Sum	114	InnoDB	utf8mb4_general_ci	288.0 KiB	0 B

Gambar 5.1.2 Detail Tabel *Database*

Pada Gambar 5.1.3 merupakan tabel gejala, yang digunakan untuk menyimpan data-data gejala mengenai *covid-19* yang ditambahkan oleh admin. Dalam tabel gejala terdapat atribut kodegejala (*primary key*), isigejala dan date.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	kodegejala	varchar(100)	utf8mb4_general_ci		No	None			
<input type="checkbox"/> 2	isigejala	varchar(200)	utf8mb4_general_ci		No	None			
<input type="checkbox"/> 3	date	timestamp			No	current_timestamp()	ON UPDATE CURRENT_TIMESTAMP()		

Gambar 5.1.3 Implementasi Tabel Gejala

Pada Gambar 5.1.4 merupakan tabel penyakit, yang digunakan untuk menyimpan data-data penyakit mengenai *covid-19* yang ditambahkan oleh admin. Dalam tabel penyakit terdapat atribut kodepenyakit yang merupakan (*primary key*), nama, info, solusi dan date.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	kodepenyakit	varchar(100)	utf8mb4_general_ci		No	None			
<input type="checkbox"/> 2	nama	varchar(200)	utf8mb4_general_ci		No	None			
<input type="checkbox"/> 3	info	text	utf8mb4_general_ci		No	None			
<input type="checkbox"/> 4	solusi	text	utf8mb4_general_ci		No	None			
<input type="checkbox"/> 5	date	timestamp			No	current_timestamp()	ON UPDATE CURRENT_TIMESTAMP()		

Gambar 5.1.4 Implementasi Tabel Penyakit

Pada Gambar 5.1.5 merupakan tabel rule, yang digunakan untuk menyimpan data-data rule aturan dan basis pengetahuan sesuai dengan gejala dan penyakit yang ditambahkan oleh admin. Dalam tabel rule terdapat atribut idrule (*primary key*),

kodepenyakit (*Foreign Key*), kodegejala (*Foreign Key*), kodecfpakar (*Foreign Key*), idadmin (*Foreign Key*) dan date.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	idrule	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	kodepenyakit	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
3	kodegejala	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
4	kodecfpakar	int(11)			No	None			Change Drop More
5	idadmin	int(11)			No	None			Change Drop More
6	date	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	Change Drop More

Gambar 5.1.5 Implementasi Tabel Rule / Basis Pengetahuan

Pada Gambar 5.1.6 merupakan tabel diagnosis, yang digunakan untuk menyimpan data-data jawaban mengenai gejala yang dipilih *user* dengan rentan nilai kepastian yang dipilih pada setiap gejalanya. Dalam tabel diagnosis terdapat atribut nodiag (*primary key*), iddiagnosa, user (*Foreign Key*), kodegejala (*Foreign Key*), gejala, jawab dan date.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	nodiag	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	iddiagnosa	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
3	user	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
4	kodegejala	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
5	gejala	varchar(200)	utf8mb4_general_ci		No	None			Change Drop More
6	jawab	float			No	None			Change Drop More
7	date	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	Change Drop More

Gambar 5.1.6 Implementasi Tabel Diagnosis

Pada Gambar 5.1.7 merupakan tabel hasil, yang digunakan untuk menyimpan data-data hasil diagnosis yang telah dilakukan oleh *user*. Dalam tabel hasil terdapat

atribut idhasil (*primary key*), user (*Foreign Key*), kodepenyakit (*Foreign Key*), iddiagnosa, prediksi, ncf, pros dan date.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 idhasil	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	2 user	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	3 kodepenyakit	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	4 iddiagnosa	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	5 prediksi	int(11)			No	None			Change Drop More
<input type="checkbox"/>	6 ncf	float			No	None			Change Drop More
<input type="checkbox"/>	7 pros	float			No	None			Change Drop More
<input type="checkbox"/>	8 date	timestamp			No	current_timestamp()	ON UPDATE CURRENT_TIMESTAMP()		Change Drop More

Gambar 5.1.7 Implementasi Tabel Hasil

Pada Gambar 5.1.8 merupakan tabel *certainty factor*, yang digunakan untuk menyimpan data-data rentan nilai kepastian 0-1 beserta keterangannya. Dalam tabel *certainty factor* terdapat atribut kodecfpakar (*primary key*), nilai, ket dan date.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 kodecfpakar	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 nilai	float			No	None			Change Drop More
<input type="checkbox"/>	3 ket	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	4 date	timestamp			No	current_timestamp()	ON UPDATE CURRENT_TIMESTAMP()		Change Drop More

Gambar 5.1.8 Implementasi Tabel Nilai *Certainty Factor*

Pada Gambar 5.1.9 merupakan tabel *user*, yang digunakan untuk menyimpan data-data *user* saat melakukan registrasi sistem. Data-data *user* disimpan untuk digunakan pada saat login dan informasi dalam sistem. Dalam tabel *user* terdapat atribut *username* (*primary key*), pass, pass2, nama, gender, umur, alamat dan date.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 username	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	2 pass	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	3 pass2	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	4 nama	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	5 gender	enum('Laki - Laki', 'Perempuan')	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	6 umur	int(11)			No	None			Change Drop More
<input type="checkbox"/>	7 alamat	varchar(200)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	8 date	timestamp			No	current_timestamp()	ON UPDATE CURRENT_TIMESTAMP()		Change Drop More

Gambar 5.1.9 Implementasi Tabel User

Pada Gambar 5.1.10 merupakan tabel admin, yang digunakan untuk menyimpan data-data admin digunakan untuk melakukan login sistem pada sisi admin. Dalam tabel admin terdapat atribut idadmin (*primary key*), username, password dan nama.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	idadmin	int(11)		No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	username	varchar(20)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	3	password	varchar(20)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	4	password2	varchar(100)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	5	nama	varchar(100)	latin1_swedish_ci	No	None			Change Drop More

Gambar 5.1.10 Implementasi Tabel Admin

Pada Gambar 5.1.11 merupakan tabel artikel, yang digunakan untuk menyimpan data-data artikel kesehatan mengenai *covid-19* yang ditambahkan oleh admin. Dalam tabel artikel terdapat atribut idartikel (*primary key*), judul, gambar, info, url dan date.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	idartikel	int(11)		No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	judul	varchar(100)	utf8mb4_general_ci	No	None			Change Drop More
<input type="checkbox"/>	3	gambar	varchar(100)	utf8mb4_general_ci	No	None			Change Drop More
<input type="checkbox"/>	4	info	text	utf8mb4_general_ci	No	None			Change Drop More
<input type="checkbox"/>	5	url	varchar(100)	utf8mb4_general_ci	No	None			Change Drop More
<input type="checkbox"/>	6	date	timestamp		No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	Change Drop More

Gambar 5.1.11 Implementasi Tabel Artikel

5.1.2 Implementasi Kode Program

Implementasi beberapa potongan kode program alur proses kerja yang ada dalam sistem sesuai dengan perancangan yang telah dilakukan sebelumnya. Sistem dibuat menggunakan Bahasa pemrograman *PHP*, *HTML*, *Framerok CSS Bootstrap* dan *Javascript*. Implementasi kode program berdasarkan proses analisis dan perancangan dijelaskan secara detail, sebagai berikut :

A. Admin

Pada sisi admin terdapat beberapa fitur yang dapat digunakan untuk melakukan manajemen data dalam sistem agar sistem yang dibuat dapat berjalan secara dinamis, sebagai berikut :

1. Memanajemen Data Gejala

Dalam manajemen data gejala terdapat beberapa bagian yaitu menambahkan, menampilkan, mengubah dan menghapus data-data gejala dalam sistem.

a. Menambahkan Data Gejala

Dalam proses menambahkan data gejala menampung nilai yang didapatkan dari `$_POST` form dalam variabel. Kemudian dilakukan pengecekan terhadap kode gejala yang diinputkan, jika sudah ada maka akan keluar *alert* dan proses *insert* data tidak dilakukan. Jika tidak ada gejala yang sama maka proses *insert* akan dilakukan dengan menggunakan *query insert* data dalam *database*, sebagai berikut :

```
$kode = ($_POST["kode"]);
$gejala = ($_POST["gejala"]);
date_default_timezone_set('Asia/Jakarta');
$tgl = date("Y-m-d H:i:s");
$cekkode = mysqli_query($conn, "SELECT kodegejala FROM gejala
WHERE kodegejala = '$kode'");
if(mysqli_fetch_assoc($cekkode))
{
    echo
    "<script type='text/javascript'>
        alert('Kode Gejala Sudah Ada');
        history.back(self);
    </script>";
    return false;
}

$query="INSERT INTO `gejala`(`kodegejala`, `isigejala`, `date`)
VALUES ('$kode','$gejala','$tgl)";
mysqli_query($conn,$query);
```

b. Menampilkan Data Gejala

Dalam proses menampilkan data gejala data ditampilkan dalam bentuk tabel dengan *javascript*. Terdapat *query select* untuk menampilkan segala isi data pada tabel

gejala dengan ketentuan kode gejala disusun *ascending*. Kemudian dilakukan foreach terhadap query select untuk melakukan perulangan terhadap isi tabel untuk menampilkan keseluruhan data. Terdapat dua button untuk melakukan proses menghapus dan mengubah data berdasarkan id yang dipilih, sebagai berikut :

```
$gejala = query("SELECT * FROM gejala ORDER BY kodegejala ASC");
<table class="table table-hover table-bordered" id="example">
  <thead>
    <tr>
      <th width="50px">No.</th>
      <th width="100px">Kode Gejala</th>
      <th width="700px">Gejala</th>
      <th width="100px">Waktu Diubah</th>
      <th width="200px">Action</th>
    </tr>
  </thead>
  <tbody>
    <?php $i=1 ?>
    <?php foreach($gejala as $row):?>

      <tr>
        <td><?= $i; ?></td>
        <td><?= $row["kodegejala"]; ?></td>
        <td><?= $row["isigejala"]; ?></td>
        <td><?= $row["date"]; ?></td>
        <td>
          <div class="p-t-10 icons">
            <a button class="btn btn-primary btn-sm" data-
toggle="tooltip" data-placement="bottom" title="Update Gejala"
            name="back" type="submit" href="editgejala.php?id=<?p
hp echo $row["kodegejala"];?>" onclick="return confirm('Apakah Geja
la Akan Diupdate?') ">
              <span class="fa fa-edit"></span>
            </button></a>
          </div>
        </td>
      </tr>
    </tbody>
</table>
```

```

        <a button class="btn btn-danger btn-sm" data-
toggle="tooltip" data-placement="bottom" title="Hapus Gejala"
        name="back" href="hapusgejala.php?id=<?= $row["kodege
jala"];?>" onclick="return confirm('Apakah Gejala Akan Dihapus?')">
        <span class="fa fa-trash"> </span>
        </button></a>

    </tr>
    <?php $i++ ?>
    <?php endforeach;?>
</tbody>
</table>
<script type="text/javascript">
    $(document).ready(function ()
    {
        $('#example').DataTable
        (
            {
                dom: 'lBfrtip',
                buttons: ['copy', 'csv', 'excel', 'pdf', 'print']
            }
        );
    });
</script>
<script>
    $(document).ready(function(){
        $('[data-toggle="tooltip"]').tooltip();
    });
</script>

```

c. Mengubah Data Gejala

Dalam proses mengubah data gejala menampung nilai yang didapatkan dari parameter \$data berdasarkan array tabel. Kemudian akan dilakukan proses *update* data berdasarkan data yang dirubah berdasarkan id dipilih. Jika berhasil maka akan muncul *alert* berhasil diupdate, sebagai berikut :


```

function editgejala($data)
{
    $id          =$data["id"];
    $isigejala   =$data["gejala"];
    $kodegejala  =$data["kode"];
    $query="UPDATE gejala SET
            kodegejala = '$kodegejala',
            isigejala = '$isigejala'
            WHERE kodegejala = '$id' ";
    mysqli_query($conn,$query);
    echo "
        <script>
            alert('Gejala Berhasil Diupdate');
            window.location = 'viewgejala.php';
        </script>
    ";
}

```

d. Menghapus Data Gejala

Dalam proses menghapus data gejala pada *database* menggunakan *query delete*. Menjalankan fungsi berdasarkan parameter \$id yang mengembalikan nilai id pada data. Dilakukan proses *delete* data berdasarkan data yang dihapus berdasarkan id dipilih, sebagai berikut :

```

function hapusgejala($id)
{
    global $conn;
    mysqli_query($conn, "DELETE FROM gejala WHERE kodegejala = '$
id'");
    return mysqli_affected_rows($conn);
}

```

2. Memanajemen Data Penyakit

Dalam manajemen data penyakit terdapat beberapa bagian yaitu menambahkan, menampilkan, mengubah dan menghapus data-data penyakit dalam sistem.

a. Menambahkan Data Penyakit

Dalam proses menambahkan data penyakit menampung nilai yang didapatkan dari `$_POST` pada form dalam variabel. Kemudian dilakukan pengecekan terhadap kode penyakit yang diinputkan, jika sudah ada maka akan keluar *alert* dan proses *insert* data tidak dilakukan. Jika tidak ada penyakit yang sama maka proses *insert* akan dilakukan dengan menggunakan *query insert* data, sebagai berikut :

```
$kode = ($_POST["kode"]);
$nama = ($_POST["nama"]);
$info = ($_POST["info"]);
$solusi = ($_POST["solusi"]);
date_default_timezone_set('Asia/Jakarta');
$tgl = date("Y-m-d H:i:s");
    $cekkode = mysqli_query($conn, "SELECT kodepenyakit FROM pen
yakit WHERE kodepenyakit = '$kode'");
    if(mysqli_fetch_assoc($cekkode)
    {
        echo
        "<script type='text/javascript'>
            alert('Kode Penyakit Sudah Ada');
            history.back(self);
        </script>";
        return false;
    }
    $query="INSERT INTO `penyakit`(`kodepenyakit`, `nama`, `info`, `s
olusi`, `date`) VALUES ('$kode','$nama','$info','$solusi','$tgl)";
    mysqli_query($conn,$query);
```

b. Menampilkan Data Penyakit

Dalam proses menampilkan data penyakit data ditampilkan dalam bentuk tabel dengan *javascript*. Terdapat *query select* untuk menampilkan segala isi data pada tabel penyakit dengan ketentuan kode penyakit disusun *ascending*. Kemudian dilakukan *foreach* terhadap *query select* untuk melakukan perulangan terhadap isi tabel untuk menampilkan keseluruhan data. Terdapat dua button untuk melakukan proses menghapus dan mengubah data pada daftar penyakit berdasarkan id yang dipilih, sebagai berikut :

```
<table class="table table-hover table-bordered" id="example">
  <thead>
    <tr>
      <th width="50px">No.</th>
      <th width="50px">Kode Penyakit</th>
      <th width="100px">Penyakit</th>
      <th width="500px">Info Penyakit</th>
      <th width="900px">Solusi</th>
      <th width="200px">Waktu Diubah</th>
      <th width="400px">Action</th>
    </tr>
  </thead>
  <tbody>
    <?php $i=1 ?>
    <?php foreach($penyakit as $row):?>
      <tr>
        <td><?= $i; ?></td>
        <td><?= $row["kodepenyakit"]; ?></td>
        <td><?= $row["nama"]; ?></td>
        <td><?= $row["info"]; ?></td>
        <td><?= $row["solusi"]; ?></td>
        <td><?= $row["date"]; ?></td>
        <td>
          <div class="p-t-10 icons">
```

```

        <a button class="btn btn-primary btn-sm" data-
toggle="tooltip" data-placement="bottom" title="Update Penyakit"
        name="back" type="submit" href="editpenyakit.php?id=<
?php echo $row["kodepenyakit"];?>" onclick="return confirm('Apakah
Penyakit Akan Diupdate?') ">
        <span class="fa fa-edit"> </span>
    </button></a>

        <a button class="btn btn-danger btn-sm" data-
toggle="tooltip" data-placement="bottom" title="Hapus Penyakit"
        name="back" type="submit" href="hapuspenyakit.php?id=
<?= $row["kodepenyakit"]; ?>" onclick="return confirm('Apakah Penya
kit Akan Dihapus?') ">
        <span class="fa fa-trash"> </span>
    </button></a>
    </td>
</tr>
<?php $i++ ?>
<?php endforeach;?>
</tbody>
</table>

```

c. Mengubah Data Penyakit

Dalam proses mengubah data penyakit menampung nilai yang didapatkan dari parameter `$data` berdasarkan array tabel. Kemudian akan dilakukan proses *update* data berdasarkan data yang dirubah berdasarkan id dipilih. Jika berhasil maka akan muncul *alert* berhasil diupdate, sebagai berikut :

```

function editpenyakit($data)
{
    $id           =$data["id"];
    $kode         =$data["kode"];
    $nama         =$data["nama"];
    $info        =$data["info"];
    $solusi       =$data["solusi"];
}

```

```

        $query="UPDATE penyakit SET
        kodepenyakit = '$kode',
        nama = '$nama',
        info = '$info',
        solusi = '$solusi'
        WHERE kodepenyakit = '$id' ";

        mysqli_query($conn,$query);
        // return mysqli_affected_rows($conn);

        echo "
            <script>
                alert('Penyakit Berhasil Diupdate');
                window.location = 'viewpenyakit.php';
            </script>
        ";
    }

```

d. Menghapus Data Penyakit

Dalam proses menghapus data penyakit pada *database* menggunakan *query delete*. Menjalankan fungsi berdasarkan parameter \$id yang mengembalikan nilai id pada data. Dilakukan proses *delete* data berdasarkan data yang dihapus berdasarkan id dipilih, sebagai berikut :

```

function hapuspenyakit($id)
{
    mysqli_query($conn, "DELETE FROM penyakit WHERE kodepenyakit = '$id'");
    return mysqli_affected_rows($conn);
}

```

3. Memanajemen Data Rule atau Basis Pengetahuan

Dalam manajemen data rule terdapat beberapa bagian yaitu menambahkan, menampilkan dan menghapus data-data rule berdasarkan penyakit dan gejala yang ada dalam sistem.

a. Menambahkan Data Rule

Dalam proses menambahkan data rule menampung nilai yang didapatkan dari `$_POST` form dalam variabel. Kemudian dilakukan pengecekan terhadap kode rule yang diinputkan, jika sudah ada maka akan keluar *alert* dan proses *insert* data tidak dilakukan. Jika tidak ada penyakit yang sama maka proses *insert* akan dilakukan dengan menggunakan *query insert* data dalam *database*. Rule yang ditambahkan bersifat dinamis berdasarkan data penyakit dan gejala yang telah ditambahkan ke dalam sistem, sebagai berikut :

```
$kg = ($_POST["rg"]);
$kp = ($_POST["rp"]);
$kc = ($_POST["rc"]);
$ia = ($_SESSION["idadmin"]);
date_default_timezone_set('Asia/Jakarta');
$tgl = date("Y-m-d H:i:s");
$cckode = mysqli_query($conn, "SELECT kodepenyakit, kodegejala FROM rule WHERE kodepenyakit = '$kp' AND kodegejala = '$kg'");
if(mysqli_fetch_assoc($cckode))
{
    echo
    "
```

```

        <script type='text/javascript'>
            alert('Rule Sudah Ada');
            history.back(self);
        </script>
";
return false;
}
mysqli_query($conn,'SET foreign_key_checks = 0');
$query = "INSERT INTO `rule`(`kodepenyakit`, `kodegejala`, `
kodecfpakar`, `idadmin`, `date`) VALUES ('$kp','$kg','$kc','$ia','$
tgl')";
mysqli_query($conn,$query);
if(mysqli_affected_rows($conn)>0)
{
    echo "
        <script>
            alert('Rule Berhasil Disimpan');
            window.location = 'knowledge.php';
        </script>
";
}

```

b. Menampilkan Data Rule

Dalam proses menampilkan data rule data ditampilkan dalam bentuk tabel dengan *javascript*. Terdapat *query select* untuk menampilkan segala isi data pada tabel rule dengan ketentuan *id* rule disusun *ascending*. Kemudian dilakukan *foreach* terhadap *query select* untuk melakukan perulangan terhadap isi tabel untuk menampilkan keseluruhan data. Terdapat button untuk melakukan proses menghapus data rule berdasarkan *id* yang dipilih, sebagai berikut :

```

<table class="table table-hover table-bordered" id="example">
  <thead>
  <tr>

```

```

<th width="50px">No.</th>
<th width="50px">Kode Penyakit</th>
<th width="700px">Penyakit</th>
<th width="50px">Kode Gejala</th>
<th width="1000px">Gejala</th>
<th >CF Pakar</th>
<th width="200px">Keterangan</th>
<th width="200px">Waktu</th>
<th width="100px">Action</th>
</tr>
</thead>
<tbody>
<?php $i=1 ?>
<?php foreach($rule as $row):?>
<tr>
<td><?= $i; ?></td>
<td><?= $row["kodepenyakit"]; ?></td>
<td><?= $row["nama"]; ?></td>
<td><?= $row["kodegejala"]; ?></td>
<td><?= $row["isigejala"]; ?></td>
<td><?= $row["nilai"]; ?></td>
<td><?= $row["ket"]; ?></td>
<td><?= $row["date"]; ?></td>
<td>
<div class="p-t-10 icons">
<a button class="btn btn-danger btn-sm" data-toggle="tooltip"
data-placement="bottom" title="Hapus Basis Pengetahuan"
name="back" type="submit" href="hapusrule.php?id=<?php echo
$row["idrule"]; ?>" onclick="return confirm('Apakah Rule Akan
Dihapus?')"><span class="fa fa-trash"></span>button</a>
</td>
</tr>
<?php $i++ ?>
<?php endforeach;?>
</tbody>
</table>

```


c. Menghapus Data Rule

Dalam proses menghapus data rule basis pengetahuan pada *database* menggunakan *query delete*. Menjalankan fungsi berdasarkan parameter \$id yang mengembalikan nilai id pada data. Dilakukan proses *delete* data berdasarkan data yang dihapus berdasarkan id dipilih, sebagai berikut :

```
function hapusrule($id)
{
    mysqli_query($conn, "DELETE FROM rule WHERE idrule = '$id'");
    return mysqli_affected_rows($conn);
}
```

B. User

Pada sisi *user* terdapat beberapa fitur yang dapat digunakan untuk melakukan proses diagnosis dalam sistem, sebagai berikut :

1. Start Diagnosis

Dalam proses *Start Diagnosis user* akan ditampilkan dengan data-data gejala yang telah ditambahkan ke dalam sistem. *User* dapat melakukan pemilihan gejala berdasarkan kepastian nilai *certainty factor*. Proses dilakukan dengan melakukan perulangan *foreach* tabel gejala dan tabel nilai *certainty factor* dalam *database* dan data disimpan dalam array, sebagai berikut :

```
<table class="table table-hover">
<thead>
<tr>
<th width="50px">No.</th>
<th width="100px">Kode Gejala</th>
<th width="700px">Gejala</th>
<th width="300px">Kepastian</th>
</tr>
</thead>
<?php $i=1 ?>
```

```

<?php foreach($gejala as $row):?>
<tbody>
<tr>
    <td><?= $i; ?></td>
    <td><?= $row["kodegejala"]; ?></td>
    <td>Apakah anda mengalami <?=$row["isigejala"];?> ?</td>
    <td><select name="pilih[]" required="required" class="custom-
select" id="inputGroupSelect01">
        <?php foreach ($cfp as $row): ?>
            <option value=<?=$row["nilai"];?>> <?=$row["ket"];?>
(<?=$row["nilai"];?>) </option>
        <?php endforeach;?></td>
</tr>
<?php $i++ ?>
<?php endforeach;?>
tbody>
</table>

```

2. *History* Diagnosis

Dalam *History* Diagnosis *user* akan ditampilkan dengan hasil riwayat diagnosis yang sudah pernah dilakukan dalam sistem. Terdapat *button detail* untuk melihat informasi tambahan mengenai riwayat diagnosis, sebagai berikut :

```

$usern = $_SESSION["username"];
$hasil = query("SELECT * FROM hasil h
INNER JOIN (SELECT nama, kodepenyakit FROM penyakit) p ON
h.kodepenyakit = p.kodepenyakit
WHERE h.user = '$usern' ORDER BY h.idhasil ASC");
<table id="example" class="table table-hover table-bordered">
<thead>
<tr>
    <th >No.</th>
    <th >ID Hasil</th>
    <th >Username</th>
    <th >Kode</th>
    <th >Penyakit</th>
    <th >Certainty Factor</th>
    <th >Persentase</th>
    <th >Waktu</th>
    <th >Action</th>
</tr>
</thead>

```

```

<tbody>
<?php $i=1 ?>
<?php foreach($hasil as $row):?>
    <tr>
        <td ><?= $i; ?></td>
        <td ><?= $row["idhasil"]; ?></td>
        <td ><?= $row["user"]; ?></td>
        <td ><?= $row["kodepenyakit"]; ?></td>
        <td ><?= $row["nama"]; ?></td>
        <td ><?= $row["ncf"]; ?></td>
        <td ><b><?= $row["pros"]; ?> %</td></b>
        <td ><?= $row["date"]; ?></td>
        <td>
<a button class="btn btn-success btn-sm" data-toggle="tooltip"
data-placement="bottom" title="Detail Riwayat Diagnosis"
        name="back" href="detail.php?id=<?=
$row["idhasil"];?>"><span class="fa fa-info-circle"></span>
</button></a>
</td>
</tr>
<?php $i++ ?>
<?php endforeach;?>
</tbody>
</table>

```

C. Proses Perhitungan Metode

Pada proses perhitungan metode akan dijelaskan alur kerja metode yang digunakan dalam penelitian yaitu *certainty factor*, sebagai berikut :

a. Menyimpan Jawaban *User*

Jawaban user akan disimpan dalam database untuk dilakukan diagnosis penyakit dan perhitungan terhadap metode. Membuat variabel *\$data* yang digunakan untuk menampung data *array* jawaban *user*. Mendeklarasikan *array \$indexArr* dan melakukan *foreach* terhadap data gejala. Kemudian melakukan *array_push* yang digunakan untuk menyisipkan *index* dalam *array*. Melakukan query insert data untuk menyimpan data jawaban *user*, sebagai berikut :

```

$diag = query("SELECT kodegejala, isigejala FROM gejala");
$data = $_POST["pilih"];
    $idx = 0;

```

```

$user = $_SESSION["username"];
$indexsArr = [];
foreach ($diag as $row)
{
    $kode = $row["kodegejala"];
    $gej = $row["isigejala"];
    $jw = $data[$idx];
    if (floatval($jw) != 0.)
    {
        array_push($indexsArr, $jw);
    }

    $query="INSERT INTO `diagnosa`(`iddiagnosa`, `user`,
`kodegejala`, `gejala`, `jawab`, `date`) VALUES
('$news', '$user', '$kode', '$gej', '$jw', '$tgl')";
    mysqli_query($conn, $query);
}

```

b. Pengecekan Jawaban Terhadap Rule Basis Pengetahuan

Melakukan pengecekan jumlah penyakit yang ada dengan *library* *mysqli_num_rows* pada *query select* penyakit. Mendeklarasikan *array* \$arrp[] untuk menampung nilai perulangan dari *array foreach*, kemudian dilakukan *array_push* untuk menyisipkan nilai kodepenyakit ke dalam *array*. Melakukan proses perulangan *for* berdasarkan jumlah penyakit dengan *query inner join* menggabungkan beberapa tabel untuk melakukan proses perhitungan metode, sebagai berikut :

```

$totalcf = [];
$ctp = "SELECT kodepenyakit FROM penyakit";
$cfc = mysqli_query($conn, $ctp);
$countp = mysqli_num_rows($cfc);

```

```

$arrp = [];
foreach ($cfc as $db)
{
    $dbb = ($db["kodepenyakit"]);

    array_push($arrp, $dbb);
}
for ($cp = 0; $cp < $countp; $cp++)
{
    $sql = "SELECT * FROM diagnosa
INNER JOIN rule ON rule.kodegejala = diagnosa.kodegejala
INNER JOIN cfpakar ON cfpakar.kodecfpakar = rule.kodecfpakar
INNER JOIN (SELECT kodepenyakit, nama FROM penyakit) p
ON p.kodepenyakit = rule.kodepenyakit
WHERE diagnosa.iddiagnosa='$news' AND rule.kodepenyakit
IN ('$arrp[$cp]')
ORDER BY diagnosa.nodiag ASC";
$dgn = mysqli_query($conn, $sql);

```

c. Perkalian Nilai Kepastian User Dengan Nilai Kepastian Pakar

Melakukan perulangan *foreach* terhadap query inner join untuk mengambil nilai kepastian dari jawaban user dan pakar berdasarkan gejala bersangkutan. Kemudian terdapat kondisi jika jawaban tidak 0, maka dilakukan proses perkalian nilai berdasarkan gejala dan dilakukan *array_push()* untuk menyisipkan nilai hasil perkalian ke dalam *array*, sebagai berikut :

```

foreach ($dgn as $d)
{
    $t = floatval($d["nilai"]);

```

```

$j = floatval($d["jawab"]);
if ($j != 0)
{
    array_push ($res, ($t * $indexsArr[$six])); // cfu * cfp
    $six += 1;
}
}

```

d. Menghitung Jumlah Jawaban Gejala yang Dipilih User

Setelah dilakukan proses penyisipan data ke dalam *array* untuk menampung jawaban dari user, maka dilakukan *count()* untuk menghitung jumlah yang dipilih user. Pada proses *count()* -1 dikarenakan *index array* dimulai dari 0, kemudian terdapat kondisi jika jumlah = 0 yang artinya user hanya memilih satu gejala. Jika jumlah = 1 maka jumlah jawaban dari user adalah dua gejala, sehingga dilakukan proses perhitungan menggunakan *CFCombine* karena sudah lebih dari satu gejala. Jika jumlah > 1 maka jumlah jawaban user adalah lebih dari dua gejala, maka proses perhitungan menggunakan perulangan *for* dan dilakukan *sorting* nilai terbesar, sebagai berikut :

```

$len = count($res) - 1;
$hasilcf = 0.;
if ($len == 0)
{
    $hasilcf = $res[0];
}
else if ($len == 1)
{
    $res_akhir = $res[$len];
    $hasilcf = $res[0] + $res_akhir * (1 - $res[0]);
}
else if ($len > 1)
{
    $hasilcf = $res[0] + $res[1] * (1 - $res[0]);
    for ($i = 1; $i < $len; $i++)
    {

```

```

        $hasilcf = $hasilcf + $res[$i+1] * (1 - $hasilcf);
    }
    $totalcf[$cp] = $hasilcf;
}
rsort($totalcf);

```

e. Penentuan Diagnosis Penyakit

Setelah dilakukan perhitungan terhadap gejala yang dipilih dan berdasarkan jumlah penyakit yang ada. Selanjutnya adalah menentukan diagnosis penyakit dengan membandingkan dengan basis pengetahuan. Kemudian dilakukan sorting terhadap data terbesar yang menjadi hasil dari diagnosis, sebagai berikut :

```

foreach ($ruleQuery as $rule)
{
    $status = 0;
    foreach ($dgQuery as $diagnosa)
    {
        if ($rule["kodegejala"] == $diagnosa["kodegejala"])
        {
            $penyakit = $rule["kodepenyakit"];
            $tempData[$idx] = $penyakit;
            $status = 1;
        }
    }
    if($status == 1)
    {
        $idx += 1;
    }
}
foreach ($penyakitQuery as $py)
{
    $p = $py["kodepenyakit"];
    $resultData[$idx][0] = $p;
    $inc = 0;
    for ($i = 0; $i < count($tempData); $i++)

```

```

    {
        $data = $tempData[$i];

        if ($p == $data)
        {
            //perhitungan
            $inc += 1;
        }
    }
    $resultData[$idx][1] = $inc;
    $idx += 1;
}
array_multisort(array_map(function($index)
{
    return $index[1];
},
$resultData), SORT_DESC, $resultData);

```

f. Menampilkan Hasil Diagnosis

Setelah selesai melakukan perhitungan metode maka data-data hasil diagnosis akan disimpan dalam database dengan query *insert*. Persentase yang didapatkan merupakan hasil perkalian dari nilai *certainty factor* terbesar dikalikan 100%. Kemudian data akan muncul dalam *history* atau riwayat diagnosis dari *user* bersangkutan yang telah melakukan diagnosis, sebagai berikut :

```

<table class="table table-bordered table-hover">
    <?php $i=1 ?>
        <?php $cp=0 ?>
        <?php foreach($resultData as $hasil) : ?>
            <tbody>
                <tr>
                    <td><?=$i?></td>
                    <?php $hx = query("SELECT * FROM penyakit WHERE kodepenyakit
= '$hasil[0]'" ) [0];?>
                    <td><?= $hasil[0]; ?></td>

```

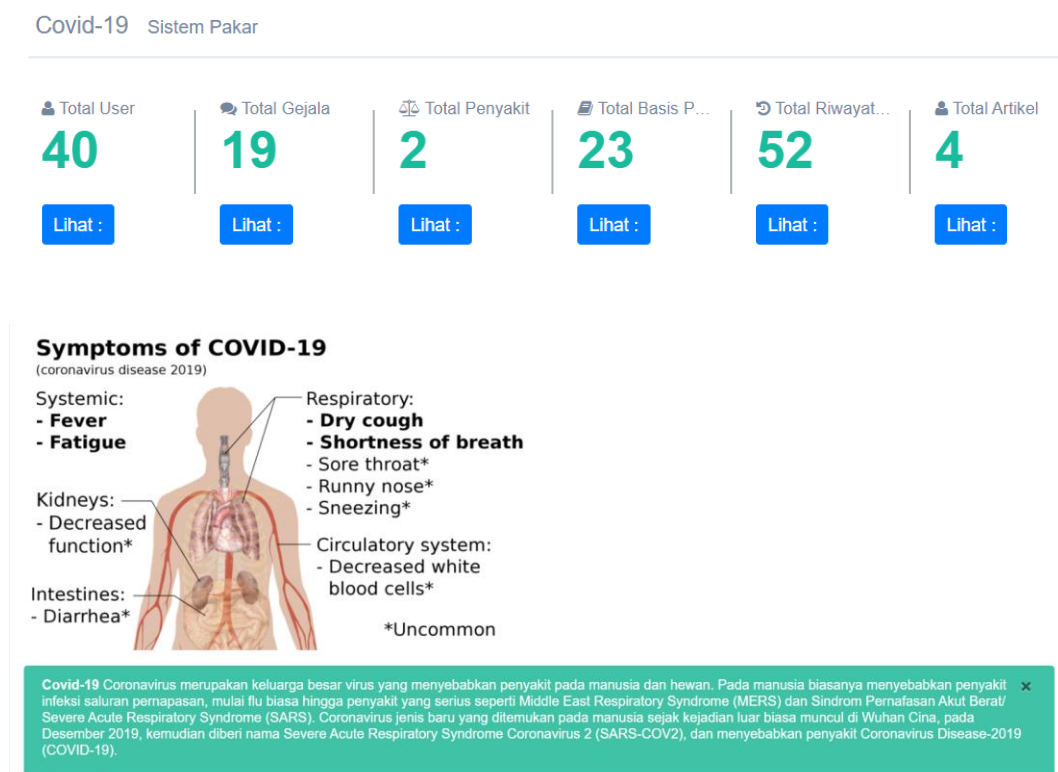


```
        <td><?= $hx["nama"]; ?></td>
        <td><?= $totalcf[$cp]; ?></td>
    </tr>
</tbody>
<?php $i++ ?>
<?php $cp++ ?>
    <?php $pros = $totalcf[0]*100;?>
    <?php $query="INSERT INTO `hasil`(`idhasil`, `user`,
`kodepenyakit`, `iddiagnosa`, `prediksi`, `ncf`, `pros`, `date`)
VALUES ('$idhasil','$user','$hasil[0]', '$news', '$hasil[1]',
'$totalcf[0]', '$pros', '$tgl')";
    mysqli_query($conn,$query);?>
<?php endforeach;?>
```

5.1.3 Implementasi Tampilan Sistem

Implementasi tampilan *user interface* dari sistem sesuai dengan perancangan desain tampilan yang dilakukan sebelumnya, sebagai berikut :

Tampilan dashboard pada admin yang merupakan halaman utama saat selesai melakukan login terdapat pada gambar 5.1.12, menampilkan jumlah data mengenai data user, gejala, penyakit, basis pengetahuan, riwayat diagnosis dan artikel yang telah ditambahkan dalam sistem. Jumlah data tersebut bersifat dinamis mengikuti penyesuaian data yang ditambahkan ke dalam sistem. Terdapat *button* lihat untuk melihat setiap detail data yang dimuat pada dashboard. Memuat gambar umum mengenai beberapa gejala atau *symptoms covid-19*, serta informasi umum mengenai *covid-19* berdasarkan data *World Health Organization (WHO)*.



Gambar 5.1.12 Implementasi Tampilan *Dashboard Admin*

Tampilan untuk manajemen data gejala terdapat pada gambar 5.1.13 menampilkan daftar gejala-gejala yang telah ditambahkan dalam sistem. Setiap data memuat kode gejala, keterangan gejala, waktu diubah dan aksi. Pada data gejala dapat dilakukan *update*, *delete*, *searching*, *sorting*, *pagination* dan *exporting* data.





No.	Kode Gejala	Gejala	Waktu Diubah	Action
1	GC01	Flu atau Pilek (Hidung Tersumbat)	2021-05-29 11:45:50	[Edit] [Delete] [X]
2	GC02	Berisik - Berisik	2021-03-11 12:28:01	[Edit] [Delete] [X]
3	GC03	Demam > 38 Derajat Celcius	2021-03-11 12:28:09	[Edit] [Delete] [X]
4	GC04	Batuk Kering	2021-03-11 12:28:16	[Edit] [Delete] [X]
5	GC05	Riwayat Kontak Erat dengan Kasus Probable atau Konfirmasi	2021-05-29 11:45:09	[Edit] [Delete] [X]
6	GC06	Sesak Nafas	2021-03-11 12:28:43	[Edit] [Delete] [X]
7	GC07	Sakit Tenggorokan	2021-03-11 12:28:50	[Edit] [Delete] [X]
8	GC08	Nyeri Dada	2021-03-11 12:28:58	[Edit] [Delete] [X]
9	GC09	Nyeri Otot atau Kelelahan	2021-03-11 12:29:08	[Edit] [Delete] [X]
10	GC10	Anosmia atau Berkurangnya Indra Penciuman	2021-03-11 12:29:17	[Edit] [Delete] [X]

Gambar 5.1.13 Implementasi Tampilan Memanajemen Data Gejala

Tampilan untuk menambahkan data gejala terdapat pada gambar 5.1.14 menampilkan form untuk menginputkan data gejala-gejala ke dalam sistem. Terdapat dua form yaitu form kodegejala dan form keterangan gejala.

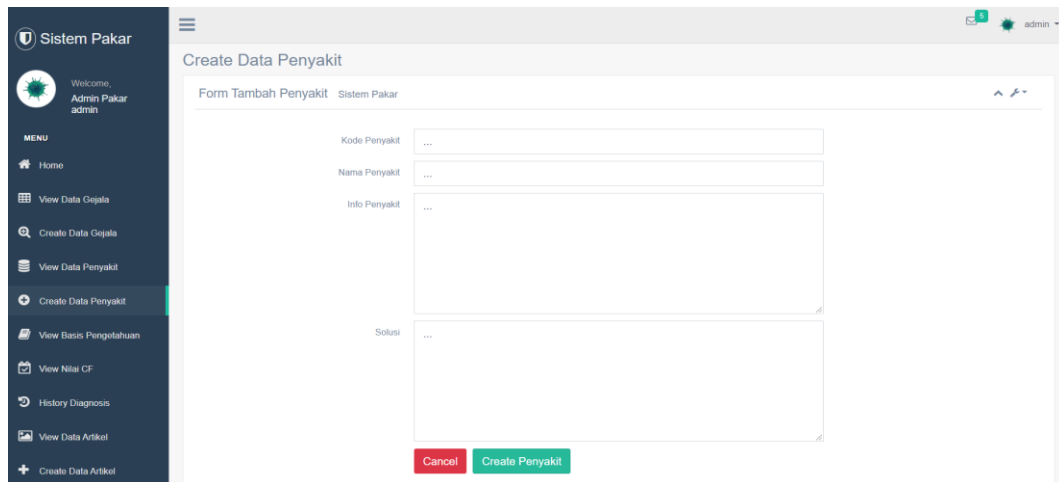
Gambar 5.1.14 Implementasi Tampilan Menambahkan Data Gejala

Tampilan untuk manajemen data penyakit terdapat pada gambar 5.1.15 menampilkan daftar penyakit-penyakit yang telah ditambahkan dalam sistem. Setiap data memuat kode penyakit, keterangan penyakit, Informasi penyakit, Solusi, waktu diubah dan aksi. Pada data penyakit dapat dilakukan *update*, *delete*, *searching*, *sorting*, *pagination* dan *exporting* data.

No.	Kode Penyakit	Penyakit	Info Penyakit	Solusi	Waktu Diubah	Action
1	PC01	Suspek Corona Virus Disease (Covid-19)	Mengalami salah satu atau beberapa gejala infeksi saluran pernapasan (ISPA), seperti demam atau riwayat demam dengan suhu di atas 38 derajat Celsius dan salah satu gejala penyakit pernapasan, seperti batuk, sesak napas, sakit tenggorokan, bersin-bersin dan pilek	Makan makanan bergizi, Rajin olahraga dan istirahat. Jaga kebersihan lingkungan. Tidak merokok. Minum air putih 8 gelas/hari, Makan makanan yang dimasak sempurna, Lakukan Aktifitas di Rumah dan Menerapkan 5M : Memakai masker, Mencuci tangan pakai sabun dan air mengalir, Menjaga jarak, Mengikuti kerumunan, serta Membatasi interaksi sosial.	2021-05-03 14:38:38	 
2	PC02	Probable Corona Virus Disease (Covid-19)	Mengalami gejala pematangan ISPA berat, namun belum ada hasil pemeriksaan Swab RT-PCR (Reverse transcription-Polymerase Chain Reaction) yang memastikan bahwa dirinya positif COVID-19.	Isolasi Mandiri 10 hari dengan ditambah minimal 3 hari setelah tidak ada lagi menunjukkan gejala demam dan gangguan pernapasan. Istirahatlah yang cukup di rumah dan minum air yang cukup. Bisa tetap merasa tidak nyaman, keluhan berlanjut, atau disertai dengan kesulitan bernapas (sesak atau napas cepat), segera memindahkan diri ke fasilitas pelayanan kesehatan (fasyankes). Fasyankes akan melakukan screening pasien dalam pengawasan COVID-19. Jika memenuhi kriteria pasien dalam pengawasan COVID-19, maka Anda akan dirujuk ke salah satu rumah sakit (RS) rujukan.	2021-05-04 09:17:03	 

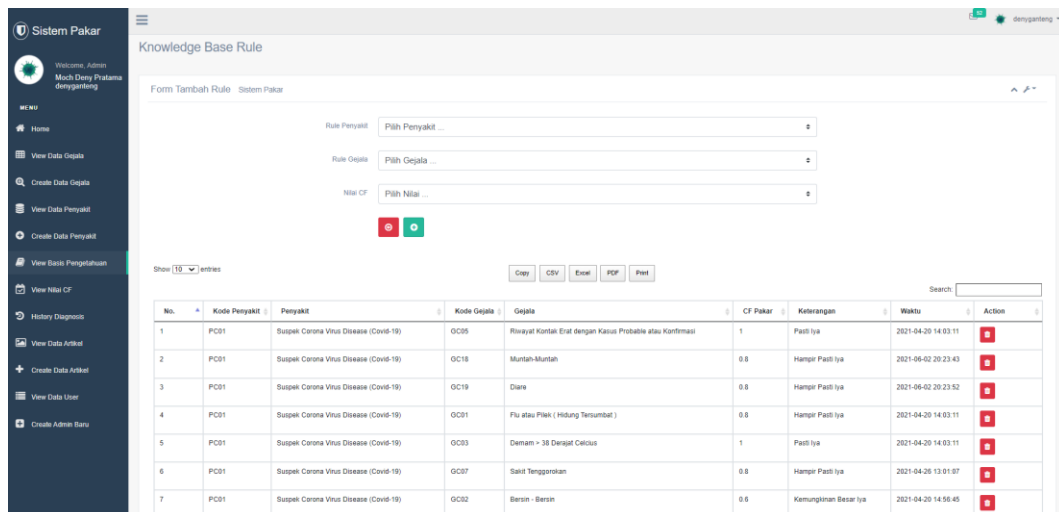
Gambar 5.1.15 Implementasi Tampilan Memanajemen Data Penyakit

Tampilan untuk menambahkan data penyakit terdapat pada gambar 5.1.16 menampilkan form untuk menginputkan data penyakit-penyakit ke dalam sistem. Terdapat empat form yaitu form kodepenyakit, form nama penyakit, informasi penyakit dan solusi yang ditawarkan jika terdiagnosis penyakit.



Gambar 5.1.16 Implementasi Tampilan Menambahkan Data Penyakit

Tampilan untuk manajemen data rule basis pengetahuan terdapat pada gambar 5.1.17 menampilkan daftar aturan-aturan yang telah ditambahkan dalam sistem. Data rule dapat ditambahkan secara dinamis berdasarkan data penyakit dan data gejala yang berelasi. Setiap data memuat kode penyakit, keterangan penyakit, kode gejala, keterangan gejala, nilai kepastian *certainty factor* pakar, keterangan *certainty factor*, waktu diubah dan aksi. Pada data rule dapat dilakukan *delete*, *searching*, *sorting*, *pagination* dan *exporting* data.



Gambar 5.1.17 Implementasi Tampilan Memanajemen Basis Pengetahuan

Tampilan untuk data nilai *certainty factor* terdapat pada gambar 5.1.18 menampilkan daftar nilai dalam sistem. Setiap data memuat keterangan dan rentan nilai *certainty factor*. Dapat dilakukan update, searching, sorting, pagination dan exporting data.

No.	Keterangan	Nilai CF Pakar	Action
1	Pasti Tidak	0	CF
2	Mungkin Tidak	0.2	CF
3	Mungkin Iya	0.4	CF
4	Kemungkinan Besar Iya	0.6	CF
5	Hampir Pasti Iya	0.8	CF
6	Pasti Iya	1	CF

Gambar 5.1.18 Implementasi Tampilan Rentan *Certainty Factor*

Tampilan untuk manajemen data *history* diagnosis keseluruhan *user* terdapat pada gambar 5.1.19 menampilkan daftar riwayat- riwayat diagnosis *user* yang telah dilakukan dalam sistem. Setiap data memuat *id* hasil, *username*, kode penyakit, keterangan penyakit, nilai *certainty factor*, persentase, waktu diubah dan aksi. Pada data *history* dapat dilakukan melihat *detail* diagnosis *user* bersangkutan, *delete*, *searching*, *sorting*, *pagination* dan *exporting* data.

No.	ID Hasil	Username	Kode	Penyakit	Certainty Factor	Persentase	Waktu	Action
11	HSL023	devid	PC02	Probable Corona Virus Disease (Covid-19)	0.4832	48.32%	2021-05-04 18:25:44	👁️ 🗑️
12	HSL024	rakamuch	PC02	Probable Corona Virus Disease (Covid-19)	0.24	24%	2021-05-04 18:30:59	👁️ 🗑️
13	HSL025	rizki	PC02	Probable Corona Virus Disease (Covid-19)	0.964476	96.4476%	2021-05-04 18:33:03	👁️ 🗑️
14	HSL026	oskaaditya	PC02	Probable Corona Virus Disease (Covid-19)	0.830278	83.6278%	2021-05-04 19:52:46	👁️ 🗑️
15	HSL027	ramadhantiph	PC01	Suspek Corona Virus Disease (Covid-19)	0.472	47.2%	2021-05-04 22:10:43	👁️ 🗑️
16	HSL028	fitriis	PC02	Probable Corona Virus Disease (Covid-19)	1	100%	2021-05-05 05:23:41	👁️ 🗑️
17	HSL029	fitriis	PC02	Probable Corona Virus Disease (Covid-19)	0.407296	40.7296%	2021-05-05 09:53:51	👁️ 🗑️
18	HSL030	syhadanny	PC01	Suspek Corona Virus Disease (Covid-19)	0.472	47.2%	2021-05-05 12:21:24	👁️ 🗑️

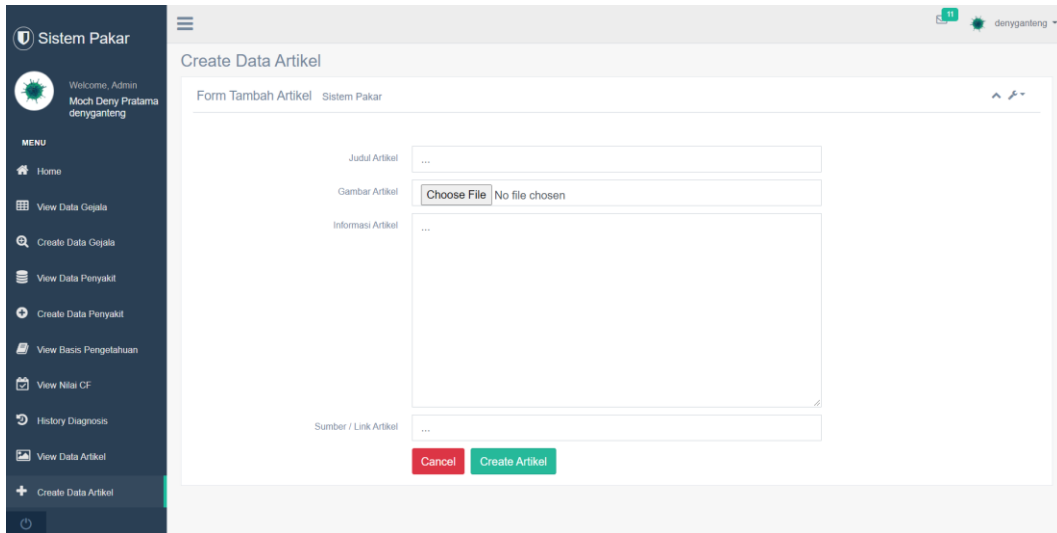
Gambar 5.1.19 Implementasi Tampilan Memanajemen *History* Diagnosis

Tampilan untuk data artikel terdapat pada gambar 5.1.20 menampilkan daftar artikel-artikel yang telah ditambahkan dalam sistem. Setiap data memuat judul artikel, gambar artikel, informasi artikel, *link* artikel dan waktu artikel ditambahkan.



Gambar 5.1.20 Implementasi Tampilan Memanajemen Data Artikel

Tampilan untuk menambahkan data artikel terdapat pada gambar 5.1.21 menampilkan form untuk menginputkan data artikel-artikel ke dalam sistem. Terdapat empat form yaitu form judul artikel, form insert gambar artikel, informasi artikel dan link dari artikel yang ditambahkan.



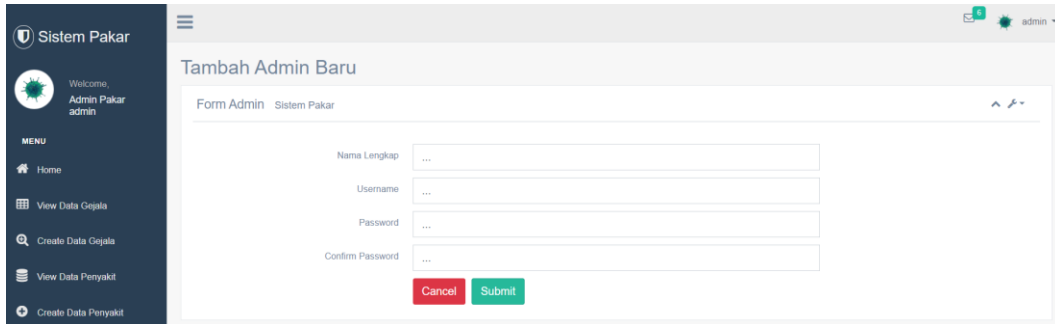
Gambar 5.1.21 Implementasi Tampilan Menambahkan Data Artikel

Tampilan untuk manajemen data *user* terdapat pada gambar 5.1.22 menampilkan daftar *user* yang telah terdaftar dalam sistem. Setiap data memuat nama, username, jenis kelamin, umur, alamat, waktu registrasi dan aksi. Pada data *user* dapat dilakukan *delete*, *searching*, *sorting*, *pagination* dan *exporting* data.

No.	Nama	Username	Password	Jenis Kelamin	Umur	Alamat	Waktu	Action
11	Syahdanny Ahmada	syahdanny	27D055ab23nc2	Laki - Laki	22	Tumpang, Malang	2021-05-05 12:15:32	[Delete]
12	Dimas Eka Adnandra	dimas capella	Dimas1590	Laki - Laki	21	Jalan gunung jati RT 05/RW 08	2021-05-05 12:30:25	[Delete]
13	Muhammad Aliyul Murtado	admin_dalub_jatin	jatin123	Laki - Laki	22	Jombang	2021-05-05 12:32:49	[Delete]
14	Harya Dwi Tama	harya123	abcd1234	Laki - Laki	21	Jalan Gunung Satu RT 10 NO.3	2021-05-05 12:41:54	[Delete]
15	J Rahma Putri	jrahmap	12345678	Perempuan	21	Jl. Ikan Beldi 1 No. 9 Blimbing-Malang	2021-05-05 12:56:48	[Delete]
16	Fian	fandjeju	salim123	Laki - Laki	23	Jl Raya PaksiKembang No. 194	2021-05-05 13:41:49	[Delete]
17	Aditya Julian Brillanzah	adityajulianb	tolongdony	Laki - Laki	21	perum buntul asri b-36	2021-05-05 14:00:48	[Delete]
18	Tsany Alwan Alauddin	alwanman	alwan123	Laki - Laki	21	Perum Taman Landungsai Indah D1	2021-05-05 14:31:47	[Delete]
19	mochamad andhika firmansyah	andhika17	Andhika17	Laki - Laki	21	Jl Swari selatan 07	2021-05-05 14:40:16	[Delete]
20	Muhammad Aif Fidaus Al-Amin	aifalamin	astaghfirullahastubillah	Laki - Laki	22	Jl. Nusa Indah IV no. 1, Jember	2021-05-05 16:22:41	[Delete]

Gambar 5.1.22 Implementasi Tampilan Memanajemen Data *User*

Tampilan untuk menambahkan data admin terdapat pada gambar 5.1.23 menampilkan form untuk menginputkan data admin baru ke dalam sistem. Terdapat empat form yaitu nama lengkap, *username*, *password* dan *confirm password*.



The screenshot shows a web interface for 'Sistem Pakar' with a dark sidebar menu. The main content area is titled 'Tambah Admin Baru' and contains a 'Form Admin' with the following fields: 'Nama Lengkap', 'Username', 'Password', and 'Confirm Password'. Each field has a placeholder '...'. Below the fields are two buttons: a red 'Cancel' button and a green 'Submit' button. The top right corner of the interface shows a user profile for 'admin'.

Gambar 5.1.23 Implementasi Tampilan Menambahkan Admin

Tampilan untuk daftar artikel kesehatan terdapat pada gambar 5.1.24 menampilkan daftar artikel-artikel yang telah ditambahkan dalam sistem. Setiap data memuat judul artikel, gambar artikel, informasi artikel, *link* artikel dan waktu artikel ditambahkan. Artikel dapat diakses oleh seluruh *user* untuk memperoleh informasi mengenai artikel yang sudah ditambahkan oleh admin.



Gambar 5.1.24 Implementasi Tampilan Artikel Kesehatan

Tampilan untuk melakukan diagnosis terdapat pada gambar 5.1.25 ditampilkan form deretan daftar gejala-gejala yang telah ditambahkan dalam sistem. Terdapat pilihan untuk rentan nilai kepastian (*certainty factor*) pada setiap gejalanya. User dapat mengisi gejala dengan kepastian yang dipilihnya, kemudian menekan *button* diagnosis.

Sistem Pakar Diagnosis Corona Virus Disease Menggunakan Metode Certainty Factor

Sesi Konsultasi Sistem Pakar

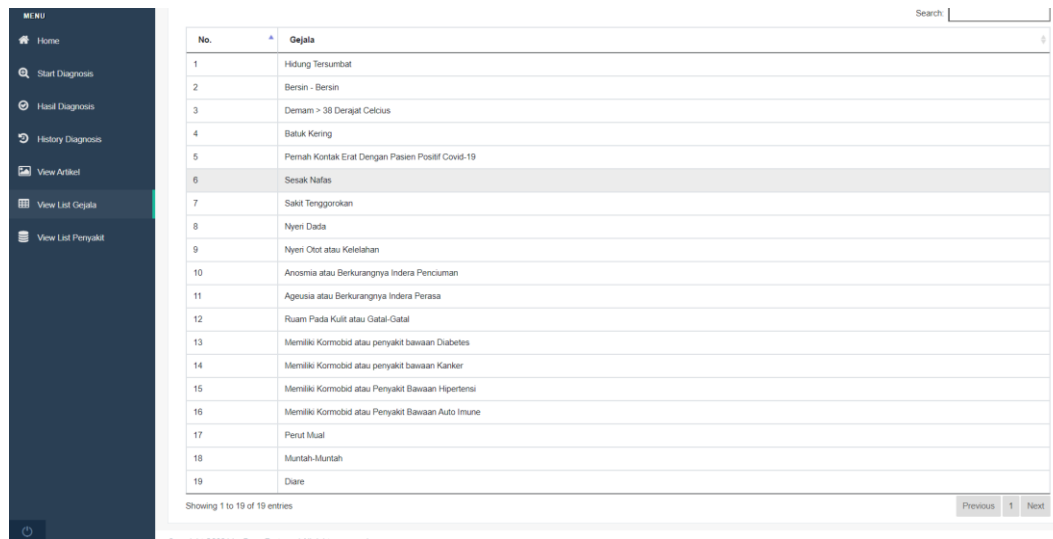
No.	Kode Gejala	Gejala	Kepastian
1	GC01	Apakah anda mengalami Flu atau Pilek (Hidung Tersumbat) ?	Pasti Tidak (0)
2	GC02	Apakah anda mengalami Bersin - Bersin ?	Pasti Tidak (0)
3	GC03	Apakah anda mengalami Demam > 38 Derajat Celcius ?	Pasti Tidak (0)
4	GC04	Apakah anda mengalami Batuk Kering ?	Pasti Tidak (0)
5	GC05	Apakah anda mengalami Riwayat Kontak Erat dengan Kasus Probable atau Konfirmasi ?	Pasti Tidak (0)
6	GC06	Apakah anda mengalami Sesak Nafas ?	Pasti Tidak (0)
7	GC07	Apakah anda mengalami Sakit Tenggorokan ?	Pasti Tidak (0)
8	GC08	Apakah anda mengalami Nyeri Dada ?	Pasti Tidak (0)
9	GC09	Apakah anda mengalami Nyeri Otot atau Kelelahan ?	Pasti Tidak (0)
10	GC10	Apakah anda mengalami Anosmia atau Berkurangnya Indera Penciuman ?	Pasti Tidak (0)
11	GC11	Apakah anda mengalami Ageusia atau Berkurangnya Indera Perasa ?	Pasti Tidak (0)
12	GC12	Apakah anda mengalami Ruam Pada Kulit atau Gatal-Gatal ?	Pasti Tidak (0)
13	GC13	Apakah anda mengalami Memiliki Kormobid atau penyakit bawaan Diabetes ?	Pasti Tidak (0)
14	GC14	Apakah anda mengalami Memiliki Kormobid atau penyakit bawaan Kanker ?	Pasti Tidak (0)
15	GC15	Apakah anda mengalami Memiliki Kormobid atau Penyakit Bawaan Hipertensi ?	Pasti Tidak (0)
16	GC16	Apakah anda mengalami Memiliki Kormobid atau Penyakit Bawaan Auto Imune ?	Pasti Tidak (0)
17	GC17	Apakah anda mengalami Perut Mual ?	Pasti Tidak (0)
18	GC18	Apakah anda mengalami Muntah-Muntah ?	Pasti Tidak (0)
19	GC19	Apakah anda mengalami Diare ?	Pasti Tidak (0)

Cancel Diagnosis

Copyright ©2021 by Dery Pratama | All rights reserved

Gambar 5.1.25 Implementasi Tampilan Konsultasi Diagnosis

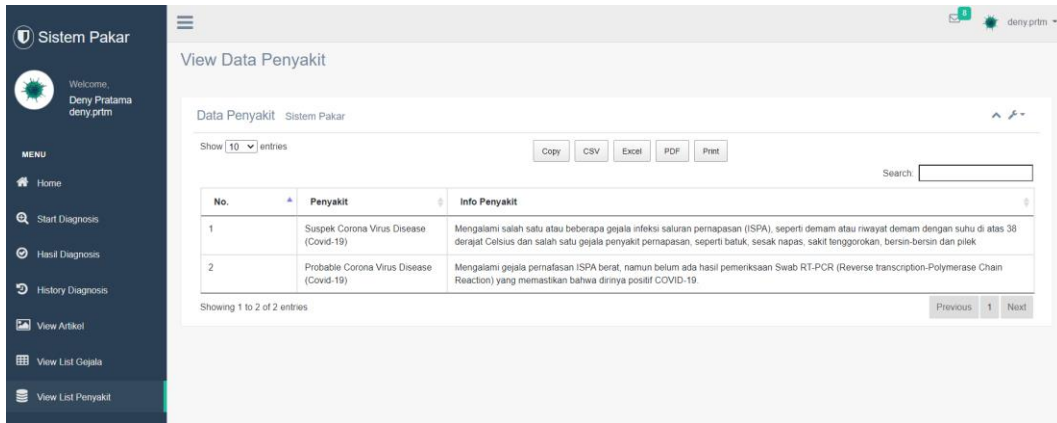
Tampilan pada *user* untuk melihat daftar gejala terdapat pada gambar 5.1.26. Pada data ini ditampilkan deretan daftar gejala-gejala yang telah ditambahkan oleh admin dalam sistem. Digunakan sebagai informasi kepada *user* mengenai gejala-gejala yang dimuat dalam sistem.



No.	Gejala
1	Hidung Tersumbat
2	Bersin - Bersin
3	Demam > 38 Derajat Celcius
4	Batuk Kering
5	Pemah Kontak Erat Dengan Pasien Positif Covid-19
6	Sesak Nafas
7	Sakit Tenggorokan
8	Nyeri Dada
9	Nyeri Otot atau Kelelahan
10	Anosmia atau Berkurangnya Indera Penciuman
11	Ageusia atau Berkurangnya Indera Perasa
12	Ruam Pada Kulit atau Gatal-Gatal
13	Memiliki Kormobid atau penyakit bawaan Diabetes
14	Memiliki Kormobid atau penyakit bawaan Kanker
15	Memiliki Kormobid atau Penyakit Bawaan Hipertensi
16	Memiliki Kormobid atau Penyakit Bawaan Auto Imune
17	Perut Mual
18	Muntah-Muntah
19	Diare

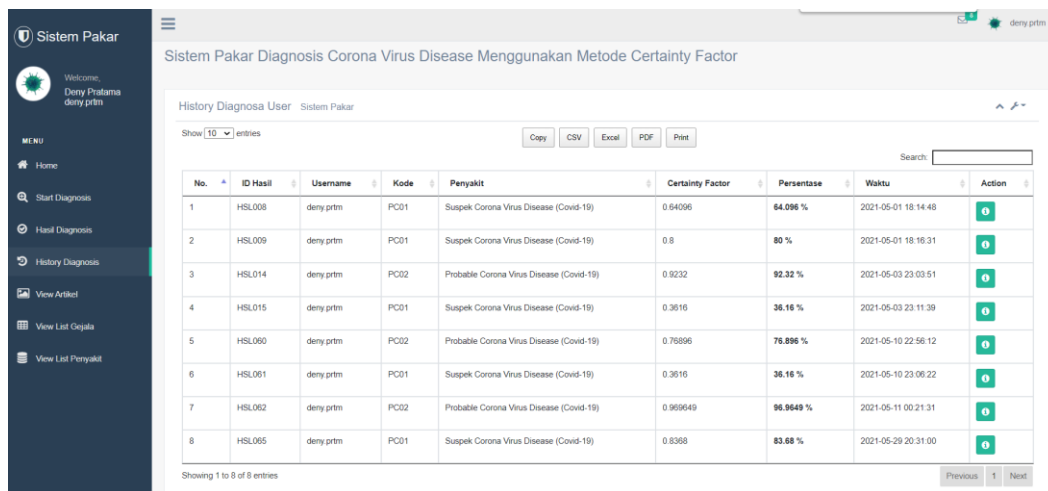
Gambar 5.1.26 Implementasi Tampilan *List* Gejala

Tampilan pada *user* untuk melihat daftar penyakit terdapat pada gambar 5.1.27. Pada data ini ditampilkan deretan daftar penyakit-penyakit yang telah ditambahkan oleh admin dalam sistem. Digunakan sebagai informasi kepada *user* mengenai penyakit-penyakit yang dimuat dalam sistem.



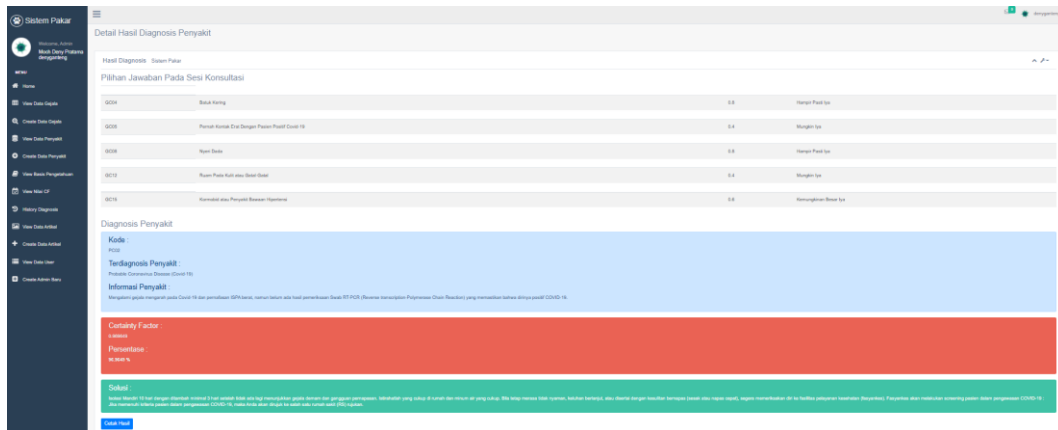
Gambar 5.1.27 Implementasi Tampilan *List Penyakit*

Tampilan untuk melihat *history* diagnosis yang telah dilakukan oleh *user* bersangkutan terdapat pada gambar 5.1.28 menampilkan daftar riwayat- riwayat diagnosis user tersebut yang telah dilakukan dalam sistem. Setiap data memuat *id* hasil, *username*, kode penyakit, keterangan penyakit, nilai *certainty factor*, persentase, waktu diubah dan aksi. Pada data *history* dapat dilakukan melihat *detail* diagnosis user bersangkutan, *searching*, *sorting*, *pagination* dan *exporting* data.



Gambar 5.1.28 Implementasi Tampilan *History* Diagnosis User

Tampilan untuk melihat Hasil Diagnosis terdapat pada gambar 5.1.29 menampilkan data diri *user*, riwayat gejala yang dipilih *user* pada saat melakukan konsultasi. Pada halaman ini juga menampilkan hasil diagnosis penyakit beserta nilai kepastiannya. Terdapat informasi secara umum dan solusi yang ditawarkan mengenai penyakit yang terdiagnosis.



Gambar 5.1.29 Implementasi Tampilan Hasil Diagnosis

5.2 Pengujian

Pengujian merupakan proses untuk menentukan apakah hasil dari penelitian sudah sesuai dengan kebutuhan sistem dan berjalan sesuai lingkungan yang diinginkan. Pengujian yang dilakukan dalam penelitian yaitu, pengujian fungsional, pengujian validitas pakar dan pengujian validitas perhitungan. Pada proses pengujian dipaparkan secara detail mengenai metode pengujian, tujuan pengujian, proses pengujian serta analisis hasil pengujian.

5.2.1 Pengujian Fungsionalitas Sistem

Pengujian fungsionalitas sistem dilakukan dengan metode pengujian *black box*. Merupakan pengujian dimana menguji keseluruhan fungsional proses kerja dari sistem apakah berjalan sesuai dengan kebutuhan yang diinginkan. Fitur sistem diuji fungsionalitas dari sisi admin terdapat dalam Tabel 5.2.1, sebagai berikut :

Tabel 5.2.1 Uji Fungsional Fitur Admin

No.	Fitur	Input	Output	Hasil
1.	Validasi Login Sistem.	<i>Username</i> Benar dan <i>Password</i> Benar.	Berhasil <i>Login</i> ke dalam Sistem.	Sesuai
		<i>Username</i> Benar dan <i>Password</i> Salah.	Gagal <i>Login</i> , muncul <i>alert</i> “ <i>Password</i> Salah”.	Sesuai
		<i>Username</i> Salah dan <i>Password</i> Benar.	Gagal <i>Login</i> , muncul <i>alert</i> “ <i>Username</i> Salah”.	Sesuai
		<i>Username</i> Salah dan <i>Password</i> Salah.	Gagal <i>Login</i> , muncul <i>alert</i> “ <i>Username dan Password</i> Salah”.	Sesuai
2.	Manajemen Data Artikel yaitu menambahkan, melihat dan menghapus artikel-artikel atau poster	Menambahkan judul, gambar, informasi, <i>link</i> sumber Data Artikel.	Berhasil menambahkan data Artikel dalam Sistem.	Sesuai

	kesehatan mengenai <i>Covid-19</i> .	Menambahkan judul, gambar, informasi, <i>link</i> sumber Data Artikel sama dengan data yang ada.	Gagal menambahkan data Artikel, muncul <i>alert</i> "Artikel sudah ada".	Sesuai
		Melihat Data Artikel.	Berhasil menampilkan daftar Artikel dalam Sistem.	Sesuai
		Menghapus Data Artikel.	Berhasil menghapus data Artikel dipilih dalam Sistem.	Sesuai
3.	Manajemen Data Gejala yaitu menambahkan, melihat, mengubah dan menghapus data gejala terkait.	Menambahkan kode dan keterangan Data Gejala.	Berhasil menambahkan data Gejala dalam Sistem.	Sesuai
		Menambahkan kode dan keterangan Data Gejala sama dengan data yang sudah ada.	Gagal menambahkan data Gejala, muncul <i>alert</i> "Gejala sudah ada".	Sesuai
		Melihat Data Gejala.	Berhasil menampilkan data Gejala dalam Sistem.	Sesuai
		Mengubah kode dan keterangan Data Gejala.	Berhasil mengubah data Gejala dipilih dalam Sistem.	Sesuai
		Menghapus Data Gejala.	Berhasil menambahkan data Gejala dipilih dalam Sistem.	Sesuai
4.	Manajemen Data Penyakit yaitu menambahkan, melihat, mengubah dan	Menambahkan kode dan keterangan Data Penyakit.	Berhasil menambahkan data Penyakit dalam Sistem.	Sesuai

	menghapus data hipotesa penyakit hasil Diagnosis.	Menambahkan kode dan keterangan Data Penyakit sama dengan data yang sudah ada.	Gagal menambahkan data Penyakit, muncul <i>alert</i> "Penyakit sudah ada".	Sesuai
		Melihat Data Penyakit.	Berhasil menampilkan data Penyakit dalam Sistem.	Sesuai
		Mengubah kode dan keterangan Data Penyakit.	Berhasil mengubah data Penyakit dipilih dalam Sistem.	Sesuai
		Menghapus Data Penyakit.	Berhasil menambahkan data Penyakit dipilih dalam Sistem.	Sesuai
5.	Manajemen Data Rule yaitu menambahkan, melihat dan menghapus data rule sebagai aturan basis pengetahuan dalam sistem pakar.	Menambahkan penyakit, gejala dan nilai kepastian Data <i>Rule</i> .	Berhasil menambahkan data <i>Rule</i> dalam Sistem.	Sesuai
		Menambahkan penyakit, gejala dan nilai kepastian Data <i>Rule</i> sama dengan data yang sudah ada.	Gagal menambahkan data <i>Rule</i> , muncul <i>alert</i> " <i>Rule</i> sudah ada".	Sesuai
		Melihat Data <i>Rule</i> .	Berhasil menampilkan data <i>Rule</i> dalam Sistem.	Sesuai
		Menghapus Data <i>Rule</i> .	Berhasil mengubah data <i>Rule</i> dipilih dalam Sistem.	Sesuai
6.	Manajemen Data <i>User</i> yaitu melihat dan menghapus data <i>user</i> yang	Melihat Data <i>User</i> Terdaftar.	Berhasil menampilkan	Sesuai

	telah teregister ke dalam sistem		data <i>User</i> dalam Sistem.	
		Menghapus Data <i>User</i> .	Berhasil menghapus data <i>User</i> dipilih dalam Sistem.	Sesuai
7.	Manajemen Data Hasil Diagnosis yaitu melihat dan menghapus data hasil setiap kali ada <i>user</i> yang melakukan diagnosis.	Melihat daftar dan detail dari Data Hasil Diagnosis.	Berhasil menampilkan data dan detail Hasil Diagnosis dalam Sistem.	Sesuai
		Menghapus Data Hasil Diagnosis.	Berhasil menghapus data Hasil Diagnosis dipilih dalam Sistem.	Sesuai
8.	Melihat daftar data dari nilai kepastian (<i>Certainty Factor</i>) beserta keterangan pada setiap nilai dengan rentan nilai 0 sampai 1.	Melihat data nilai kepastian (<i>Certainty Factor</i>).	Berhasil menampilkan nilai kepastian (<i>Certainty Factor</i>).	Sesuai
9.	Melakukan penambahan terhadap akun login admin, sehingga terdapat beberapa admin yang memiliki akses ke dalam sistem untuk memajemen data.	Nama Lengkap, <i>Username</i> , <i>Password</i> dan konfirmasi <i>Password</i> .	Berhasil menambahkan data akun login Admin dalam Sistem.	Sesuai
		<i>Password</i> dan konfirmasi <i>Password</i> tidak sama	Gagal menambahkan akun, muncul <i>alert</i> "Password tidak sama".	Sesuai

Fitur sistem yang diuji fungsionalitas dari sisi *user* terdapat dalam Tabel 5.2.2 sebagai berikut :

Tabel 5.2.2 Uji Fungsional Fitur *User*

No.	Fitur	Input	Output	Hasil
1.	Validasi Registrasi Akun Sistem.	Nama Lengkap, <i>Username</i> , <i>Password</i> ,	Berhasil melakukan	Sesuai

		Konfirmasi <i>Password</i> , Jenis Kelamin, Umur dan Alamat.	registrasi data akun login Admin dalam Sistem.	
		<i>Password</i> dan Konfirmasi <i>Password</i> tidak sama	Gagal menambahkan akun, muncul alert " <i>Password</i> tidak sama".	Sesuai
2.	Validasi Login Sistem.	<i>Username</i> Benar dan <i>Password</i> Benar.	Berhasil <i>Login</i> ke dalam Sistem.	Sesuai
		<i>Username</i> Benar dan <i>Password</i> Salah.	Gagal <i>Login</i> , muncul alert " <i>Password</i> Salah".	Sesuai
		<i>Username</i> Salah dan <i>Password</i> Benar.	Gagal <i>Login</i> , muncul alert " <i>Username</i> Salah".	Sesuai
		<i>Username</i> Salah dan <i>Password</i> Salah.	Gagal <i>Login</i> , muncul alert " <i>Username</i> dan <i>Password</i> Salah".	Sesuai
3.	Melakukan Diagnosis yaitu <i>user</i> akan dihadapkan dengan beberapa gejala kemudian <i>user</i> menginputkan nilai CF pada setiap gejala yang dialaminya.	Memilih gejala dengan menginputkan nilai kepastian dengan menu <i>dropdown</i> .	Berhasil melakukan diagnosis terhadap gejala yang dipilihnya.	Sesuai
		Tidak melakukan pemilihan gejala.	Gagal melakukan diagnosis, muncul alert "pilih setidaknya satu gejala"	Sesuai

4.	Melihat Hasil Diagnosis yaitu informasi hasil Diagnosis berupa Diagnosis penyakit, gejala yang telah dipilih, nilai CF penyakit, persentase akurasi sistem dan solusi serta cara pencegahan yang didapatkan oleh <i>user</i> dalam sistem setelah <i>user</i> melakukan Diagnosis.	Menekan <i>button</i> “Diagnosis” pada saat dalam sesi konsultasi	Menampilkan data diri <i>user</i> bersangkutan, gejala-gejala yang dipilihnya, hasil diagnosis penyakit beserta nilai kepastian dan persentase <i>certainty factor</i> .	Sesuai
5.	Melihat History Diagnosis yaitu daftar hasil Diagnosis penyakit yang telah dilakukan oleh <i>user</i> .	Memilih menu history untuk melihat riwayat diagnosis <i>user</i> yang telah dilakukan dalam sistem.	Menampilkan daftar data mengenai hasil diagnosis <i>user</i> beserta waktu dan detail diagnosis.	Sesuai
6.	Melihat Data Artikel yaitu <i>user</i> akan dihadapkan dengan beberapa artikel kesehatan mengenai covid dan seputar informasinya.	Memilih menu artikel untuk melihat data-data artikel kesehatan dalam sistem.	Menampilkan data-data artikel kesehatan yang telah ditambahkan oleh admin dalam sistem.	Sesuai
7.	Melihat beberapa daftar gejala-gejala mengenai Covid-19 yang telah ditambahkan oleh admin dalam sistem.	Memilih menu gejala untuk melihat data-data gejala dalam sistem.	Menampilkan data-data gejala yang telah ditambahkan oleh admin dalam sistem.	Sesuai
8.	Melihat beberapa daftar penyakit mengenai Covid-19 yang telah ditambahkan oleh admin dalam sistem.	Memilih menu penyakit untuk melihat data-data penyakit dalam sistem.	Menampilkan data-data penyakit yang telah ditambahkan oleh admin dalam sistem.	Sesuai

Akurasi uji coba didapatkan dengan pengujian keseluruhan fungsionalitas sistem menggunakan metode *black box* sebanyak 42 sampel, sebagai berikut :

$$\sum_{i=0}^n Akurasi = \frac{42}{42} * 100 = \mathbf{100\%}$$

Jumlah akurasi $\sum_{i=0}^n Akurasi$ dihitung berdasarkan jumlah kesesuaian dan keberhasilan $\sum_{i=0}^n Kesesuaian$ dibanding dengan total sampel $\sum Total Sampel$ mengacu pada rumus yang direpresentasikan pada persamaan 3.1 terdapat dalam Bab III Uji Coba Sistem. Hasil pengujian fungsionalitas secara *black box* didapatkan kesesuaian dengan yang diinginkan terhadap segala aktivitas pengujian memiliki tingkat akurasi uji fungsionalitas sistem sebesar 100%.

5.2.2 Pengujian Validitas Pakar

Pengujian validitas pakar merupakan pengujian untuk mengetahui apakah sistem dapat berjalan sesuai dengan hasil diagnosis pakar. Dimana proses melakukan pengujian dengan membandingkan diagnosis pakar dengan hasil dari diagnosis sistem. Pada Tabel 5.2.3 dipaparkan hasil uji validitas pakar dengan sampel 8 data uji coba hasil diagnosis dalam sistem yang dibandingkan berdasarkan diagnosis pakar. Dari 8 data hasil yang diuji cobakan menampilkan perbandingan terhadap diagnosis dalam sistem dan diagnosis pakar, sebagai berikut :

Tabel 5.2.3 Uji Validitas Diagnosis Pakar

No.	Gejala Diuji Coba	Diagnosis Pakar	Diagnosis Sistem
1.	Flu atau Pilek (Hidung Tersumbat), Bersin, Demam > 38	Suspek <i>Coronavirus Disease (Covid-19)</i>	Suspek <i>Coronavirus Disease (Covid-19)</i>
2.	Batuk Kering	Suspek <i>Coronavirus Disease (Covid-19)</i>	Suspek <i>Coronavirus Disease (Covid-19)</i>
3.	Flu atau Pilek (Hidung Tersumbat), Demam > 38, Ruam Kulit, Komorbid Hipertensi	Probable <i>Coronavirus Disease (Covid-19)</i>	Probable <i>Coronavirus Disease (Covid-19)</i>
4.	Batuk Kering, Demam > 38, Riwayat Kontak Erat, Anosmia	Probable <i>Coronavirus Disease (Covid-19)</i>	Probable <i>Coronavirus Disease (Covid-19)</i>
5.	Nyeri Otot, Anosmia, Ageusia	Probable <i>Coronavirus Disease (Covid-19)</i>	Probable <i>Coronavirus Disease (Covid-19)</i>
6.	Demam > 38, Batuk Kering, Riwayat Kontak Erat, Anosmia	Probable <i>Coronavirus Disease (Covid-19)</i>	Probable <i>Coronavirus Disease (Covid-19)</i>
7.	Flu atau Pilek (Hidung Tersumbat), Bersin	Suspek <i>Coronavirus Disease (Covid-19)</i>	Suspek <i>Coronavirus Disease (Covid-19)</i>
8.	Bersin, Demam > 38, Nyeri Dada, Komorbid Hipertensi	Probable <i>Coronavirus Disease (Covid-19)</i>	Probable <i>Coronavirus Disease (Covid-19)</i>

Akurasi uji coba didapatkan dengan pengujian validitas pakar pada sistem dengan 8 sampel, sebagai berikut :

$$\sum_{i=0}^n Akurasi = \frac{8}{8} * 100 = 100\%$$

Jumlah akurasi $\sum_{i=0}^n Akurasi$ dihitung berdasarkan jumlah kesesuaian dan keberhasilan $\sum_{i=0}^n Kesesuaian$ dibanding dengan total sampel $\sum Total Sampel$ mengacu pada rumus yang direpresentasikan pada persamaan 3.1 terdapat dalam Bab III Uji Coba Sistem. Hasil pengujian validitas pakar akurasi diagnosis didapatkan kesesuaian dengan hasil diagnosis sistem dengan hasil diagnosis pakar seperti yang tertera pada Tabel 5.2.3 memiliki tingkat akurasi uji validitas pakar sebesar 100%.

5.2.3 Pengujian Validitas Perhitungan

Pengujian validitas perhitungan merupakan pengujian untuk mengetahui apakah sistem dapat menghasilkan perhitungan yang sesuai. Dimana proses melakukan pengujian dengan membandingkan diagnosis hasil sistem dengan diagnosis hasil *excel*, berdasarkan Tabel 5.2.4, 5.2.5 dan 5.2.6 dengan 8, 10 dan 15 sampel data uji menampilkan perbandingan terhadap hasil perhitungan sistem dan perhitungan *excel*, sebagai berikut :

Tabel 5.2.4 Uji Akurasi Perhitungan Metode 8 Sampel

No.	Diagnosis	Pengujian Rumus <i>Excel</i>	Pengujian Sistem
1.	Suspek <i>Coronavirus Disease (Covid-19)</i>	0.64096	0.64096
2.	Suspek <i>Coronavirus Disease (Covid-19)</i>	0.8	0.8
3.	Probable <i>Coronavirus Disease (Covid-19)</i>	<u>0.65344</u>	<u>0.76896</u>
4.	Probable <i>Coronavirus Disease (Covid-19)</i>	0.9232	0.9232
5.	Probable <i>Coronavirus Disease (Covid-19)</i>	0.9296	0.9296
6.	Probable <i>Coronavirus Disease (Covid-19)</i>	0.9232	0.9232
7.	Suspek <i>Coronavirus Disease (Covid-19)</i>	0.3616	0.3616

8.	Probable <i>Coronavirus Disease</i> (Covid-19)	0.96964864	0.96964864
----	--	-------------------	-------------------

Tabel 5.2.5 Uji Akurasi Perhitungan Metode 10 Sampel

No.	Diagnosis	Pengujian Rumus <i>Excel</i>	Pengujian Sistem
1.	Suspek <i>Coronavirus Disease</i> (Covid-19)	0.64096	0.64096
2.	Suspek <i>Coronavirus Disease</i> (Covid-19)	0.8	0.8
3.	Probable <i>Coronavirus Disease</i> (Covid-19)	<u>0.65344</u>	<u>0.76896</u>
4.	Probable <i>Coronavirus Disease</i> (Covid-19)	0.9232	0.9232
5.	Probable <i>Coronavirus Disease</i> (Covid-19)	0.9296	0.9296
6.	Probable <i>Coronavirus Disease</i> (Covid-19)	0.9232	0.9232
7.	Suspek <i>Coronavirus Disease</i> (Covid-19)	0.3616	0.3616
8.	Probable <i>Coronavirus Disease</i> (Covid-19)	0.96964864	0.96964864
9.	Suspek <i>Coronavirus Disease</i> (Covid-19)	0.733888	0.733888
10.	Probable <i>Coronavirus Disease</i> (Covid-19)	0.4832	0.4832

Tabel 5.2.6 Uji Akurasi Perhitungan Metode 15 Sampel

No.	Diagnosis	Pengujian Rumus <i>Excel</i>	Pengujian Sistem
1.	Suspek <i>Coronavirus Disease</i> (Covid-19)	0.64096	0.64096
2.	Suspek <i>Coronavirus Disease</i> (Covid-19)	0.8	0.8
3.	Probable <i>Coronavirus Disease</i> (Covid-19)	<u>0.65344</u>	<u>0.76896</u>

4.	Probable <i>Coronavirus Disease (Covid-19)</i>	0.9232	0.9232
5.	Probable <i>Coronavirus Disease (Covid-19)</i>	0.9296	0.9296
6.	Probable <i>Coronavirus Disease (Covid-19)</i>	0.9232	0.9232
7.	Suspek <i>Coronavirus Disease (Covid-19)</i>	0.3616	0.3616
8.	Probable <i>Coronavirus Disease (Covid-19)</i>	0.96964864	0.96964864
9.	Suspek <i>Coronavirus Disease (Covid-19)</i>	0.733888	0.733888
10.	Probable <i>Coronavirus Disease (Covid-19)</i>	0.4832	0.4832
11.	Suspek <i>Coronavirus Disease (Covid-19)</i>	<u>0.392</u>	<u>0.472</u>
12.	Probable <i>Coronavirus Disease (Covid-19)</i>	1	1
13.	Suspek <i>Coronavirus Disease (Covid-19)</i>	0.4832	0.4832
14.	Suspek <i>Coronavirus Disease (Covid-19)</i>	0.16	0.16
15.	Probable <i>Coronavirus Disease (Covid-19)</i>	0.4832	0.4832

Akurasi uji coba didapatkan dengan pengujian validitas perhitungan metode pada sistem dengan 8 sampel, sebagai berikut :

$$\sum_{i=0}^n Akurasi = \frac{7}{8} * 100 = \mathbf{87.5\%}$$

Akurasi uji coba didapatkan dengan pengujian validitas perhitungan metode pada sistem dengan 10 sampel, sebagai berikut :

$$\sum_{i=0}^n Akurasi = \frac{9}{10} * 100 = \mathbf{90\%}$$

Akurasi uji coba didapatkan dengan pengujian validitas perhitungan metode pada sistem dengan 15 sampel, sebagai berikut :

$$\sum_{i=0}^n Akurasi = \frac{13}{15} * 100 = \mathbf{86.66666667\%}$$

Jumlah akurasi $\sum_{i=0}^n Akurasi$ dihitung berdasarkan jumlah kesesuaian dan keberhasilan $\sum_{i=0}^n Kesesuaian$ dibanding dengan total sampel $\sum Total Sampel$ mengacu pada rumus yang direpresentasikan pada persamaan 3.1 terdapat dalam Bab III Uji Coba Sistem. Hasil pengujian validitas perhitungan didapatkan kesesuaian dengan hasil diagnosis perhitungan *excel* dengan hasil diagnosis sistem, tetapi terdapat dua data yang memiliki ketidaksesuaian hasil perhitungan sistem dengan hasil perhitungan *excel*. Nilai yang berbeda terdapat pada data no. 3 dengan perbedaan 0.65344 banding 0.76896 dan pada data no. 11 dengan perbedaan 0.392 banding 0.472, seperti yang tertera pada Tabel 5.2.6. Berdasarkan jumlah sampel uji coba memiliki tingkat akurasi uji validitas perhitungan *certainty factor* sebesar 87.5%, 90% dan 86.66666667%.