

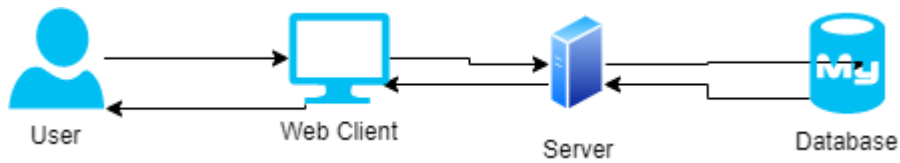
Mochamad Faizin Ahsan

Faizin110998@gmail.com

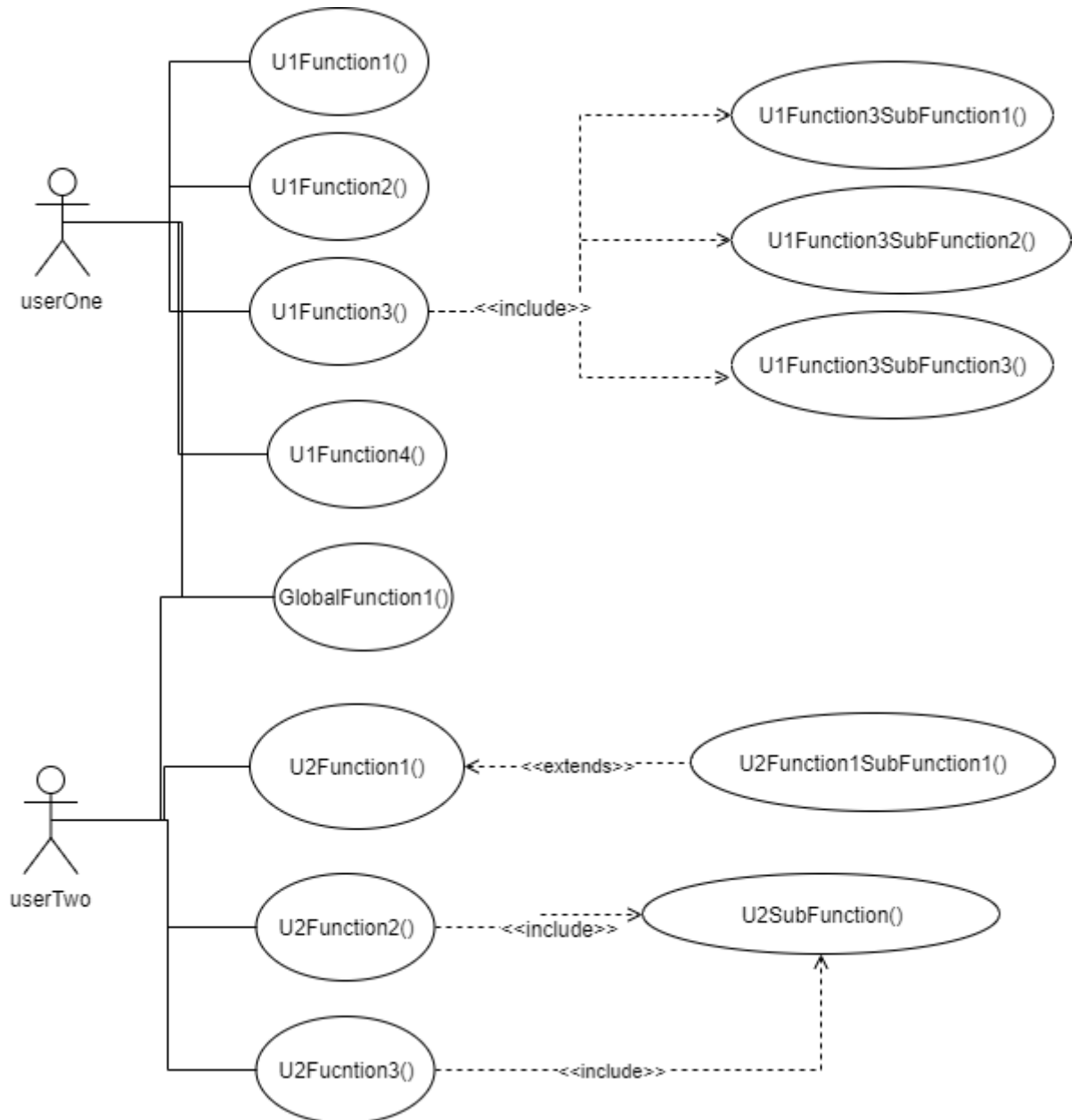
BASIC KNOWLEDGE ANSWERS

1. Penggunaan OOP dalam sebuah aplikasi web itu seperti penggunaan kelas untuk mendefinikan model dan controller yang menggunakan konsep MVC. Karena pengalaman saya membuat web menggunakan PHP Laravel, contoh penggunaan nya seperti mendefinisikan tabel menggunakan kelas lalu migrasi dengan kelas migrasi juga sehingga saat dipanggil di controller.
Bukan itu saja, namun konsep **Enskapsulasi** dapat diterapkan di pemrograman web untuk menyembunyikan informasi dari suatu data, misalnya ada beberapa data user yang sensitif yang harus di sembunyikan maka dapat menggunakan konsep enskapsulasi.
Abstraksi sering digunakan dalam pengembangan web yang menyembunyikan informasi yang terlalu detail dan hanya menampilkan informasi yang lebih misal pembuatan fungsi yang hanya bisa diakses oleh kelas” tertentu. Lalu untuk memfokuskan satu fungsi untuk satu tugas.
Inheritance mengambil sifat dari kelas parent, seperti User dengan berbagai macam jenis. Semisal kita ambil contoh database. Ketika kita memberlakukan sistem superadmin-admin-user-guest dalam hirarki pengguna web.
2. Php framework yang saya ketahui itu adalah Code Igniter(CI), Laravel, dan Lumen. Namun yang pernah saya pakai adalah Laravel dan CI. Saya pernah memaka CI untuk tugas pembuatan pustaka jurnal di web saya untuk tugas kuliah. Sedangkan Laravel pernah saya pakai saat magang di perusahaan sebelumnya untuk membuat projek mereka yaitu Hotel VIP Membersip.
3. Pertama responsive itu artinya dapat digunakan di berbagai macam kondisi layar atau device seperti layar desktop, mobile atau tablet. Untuk menangani kondisi tadi dapat dilakukan dengan berbagai macam teknologi salah satunya CSS yang dibuat dengan responsivitas seperti pada Framework css Bootstrap atau Materialize. Namun saya juga dapat memakai Framework front end yang lebih populer di kalangan developer Front End seperti React js, Vue.js, Angular.js dan lainnya. Hal ini akan menangani kondisi layar yang berbeda beda.
Namun beda jika web ingin di gunakan seperti Apps di mobile maka diperlukannya teknologi lain yaitu Progressive Web Apps(PWA). PWA adalah hal yang diperlukan untuk membuat web terasa seperti mobile dan telah di kembangkan oleh google, dan bahasa yang digunakannya adalah javascript.
4. Pertama Komponen yang perlu ada:
 - a. Server
 - i. Server ini bisa yang sudah jadi atau VPS
 - ii. Server yang di install OS berupa Linux
 - iii. Server di install Apache atau Nginx atau yang lain sesuai pilihan perusahaan
 - iv.
 - b. Pengelola dan Penyimpanan Data
 - i. Seperti MySQL, Postgresql, MondoDB dan lainnya.
 - c. Jaringan Internet

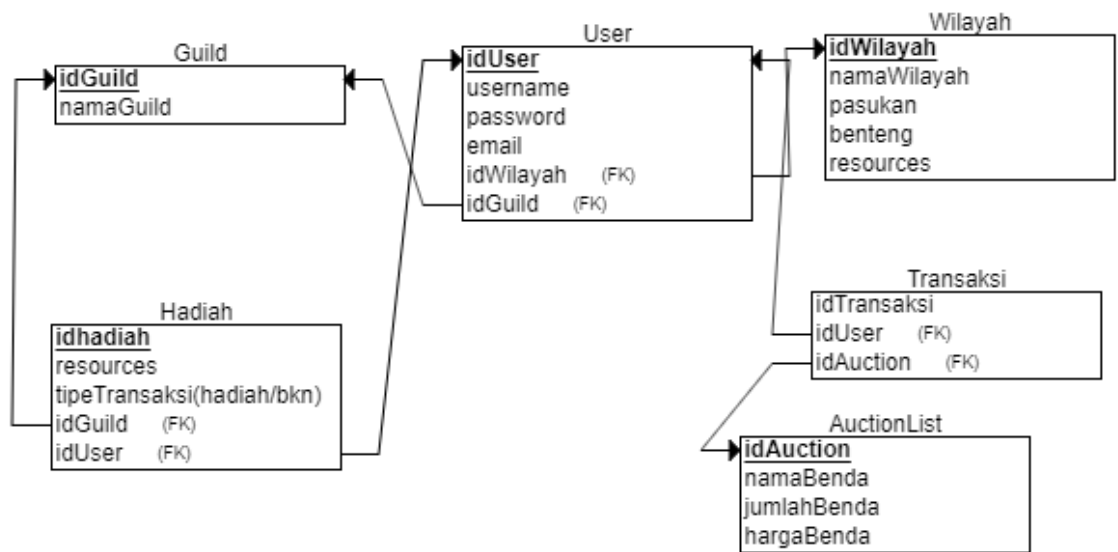
Arsitektur yang dapat dibuat tergantung kebutuhan, namun yang saya pelajari adalah arsitektur seperti pembuatan UML untuk software design dan menggunakan pendekatan object oriented. Untuk secara keseluruhan arsitektur nya seperti ini:



Namun jika menggunakan pendekatan Object Oriented seperti menggunakan UseCaseDiagram maka akan seperti:



DATABASE DESIGN



Algorithmic Test/Technical Test

1. Code:

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Soal 1</title>
</head>

<body>
  <form action="soal1.php" method="POST">
    <label for="energiTerakhirRegen">Energi Terakhir Regen</label>
    <input type="text" name="energiTerakhirRegen">
    <br>
    <label for="timestampTerakhir">Timestamp Terakhir</label>
    <input type="text" name="timestampTerakhir">
    <br>
    <label for="waktuUntukReplenish">Waktu untuk replenish</label>
    <input type="text" name="waktuUntukReplenish">
    <br>
    <button type="submit">Regen Your Energy</button>
  </form>
  <?php
    // define variables and set to empty values
    $last_energy = $last_regen_timestamp = $energy_time = "";
    $time_last_regen = array(0);
  </?php
  </body>
</html>
  
```

```

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $last_energy = test_input($_POST["energiTerakhirRegen"]);
    $last_regen_timestamp = test_input($_POST["timestampTerakhir"]);
    $energy_time = test_input($_POST["waktuUntukReplenish"]);
    regen_energy($last_energy,$last_regen_timestamp,$energy_time);
}
function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
function regen_energy($last_energy, $last_regen_timestamp, $energy_time) {
    /*Algoritma di sini*/
    $current_timestamp = time();
    array_push($time_last_regen,$current_timestamp);
    $selisih_timestamp = $current_timestamp - $last_regen_timestamp;
    $output = $last_energy + round($selisih_timestamp / $energy_time);
    /*Return some_value*/
    echo "<h3>Output Energy Regen:</h3>";
    echo $output;
    echo "<br>";
}
get_thelastime_regen($time_last_regen);
function get_thelastime_regen($time_last_regen){
    /*Return some_value*/
    echo "<h3>Last Regen:</h3>";
    echo $time_last_regen[count($time_last_regen)-1];
    echo "<br>";
}
?>
</body>
</html>

```

2. Code:

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
</head>

```

```

<body>
  <div>Timer : <h2 class="countdown" id="countdown">03:00</h2>
  </div>
  <script>
    function startTimer(duration, display) {
      var start = Date.now(),
          diff,
          minutes,
          seconds;

      function timer() {
        // get the number of seconds that have elapsed since
        // startTimer() was called
        diff = duration - (((Date.now() - start) / 1000) | 0);

        // does the same job as parseInt truncates the float
        minutes = (diff / 60) | 0;
        seconds = (diff % 60) | 0;

        minutes = minutes < 10 ? "0" + minutes : minutes;
        seconds = seconds < 10 ? "0" + seconds : seconds;

        display.textContent = minutes + ":" + seconds;

        if (diff <= 0) {
          // add one second so that the count down starts at the full
          // duration
          // example 05:00 not 04:59
          alert("finished")
          start = Date.now() + 1000;
        }
      };
      // we don't want to wait a full second before the timer starts
      timer();
      setInterval(timer, 1000);
    }
    window.onload = function () {
      var threeMinutes = 60*3,
          display = document.querySelector('#countdown');
      startTimer(threeMinutes, display);
    };
  </script>
</body>
</html>

```