

LAPORAN PRAKTIKUM

MODUL 005 **PEMROGRAMAN WEB** (MODIFIKASI PIG GAME)



Disusun oleh :

Anggi Anggraini Angel Merici
201501045
Kelas : B

Dosen

Oskar Ika Adi Nugroho.,S.T.,M.T

JURUSAN SISTEM INFORMASI
SEKOLAH TINGGI ILMU KOMPUTER “YOS SUDARSO”
PURWOKERTO
2018

BAB I

TEORI DASAR

1. Sejarah Pemenemuan Javascript.

JavaScript pertama kali diperkenalkan oleh Netscape pada tahun 1995. Pada awalnya bahasa yang sekarang disebut JavaScript ini dulunya dinamai “LiveScript” yang berfungsi sebagai bahasa sederhana untuk browser Netscape Navigator 2 yang sangat populer pada saat itu. Kemudian sejalan dengan sedang giatnya kerjasama antara Netscape dan Sun (pengembang bahasa pemrograman “Java”) pada masa itu, maka Netscape memberikan nama “JavaScript” kepada bahasa tersebut pada tanggal 4 desember 1995.

Pada saat yang bersamaan Microsoft sendiri mencoba untuk mengadaptasikan teknologi ini yang mereka sebut sebagai “Jscript” di browser milik mereka yaitu Internet Explorer 3. JavaScript sendiri merupakan modifikasi dari bahasa pemrograman C++ dengan pola penulisan yang lebih sederhana dari bahasa pemrograman C++.

2. Pengertian Javascript.

JavaScript adalah bahasa pemrograman berbasis prototipe yang berjalan disisi klien. Jika kita berbicara dalam konteks web, sederhananya, kita dapat memahami JavaScript sebagai bahasa pemrograman yang berjalan khusus untuk di browser atau halaman web agar halaman web menjadi lebih hidup. Kalau dilihat dari suku katanya terdiri dari dua suku kata, yaitu Java dan Script. Java adalah Bahasa pemrograman berorientasi objek, sedangkan Script adalah serangkaian instruksi program.

3. Kelebihan dan kekurangan Javascript.

) Kelebihan dari Javascript.

- Ukuran file kecil

Script dari javascript memiliki ukuran yang kecil sehingga ketika web yang memiliki javascript ditampilkan di browser maka akses tampilannya akan lebih cepat dibandingkan ketika browser membuka suatu web yang memiliki script java. Hal ini juga sangat berkepentingan dengan daya kerja server. Semakin kecil space suatu web yang disimpan dalam suatu server maka daya kerja server ketika di browsing oleh user di internet akan tidak terlalu berat, selain itu sifat javascript client side yang tidak perlu lagi di olah oleh server ketika browser memanggil web dari sebuah server.

- Mudah untuk dipelajari

Javascript merupakan bahasa semi pemograman yang merupakan gabungan antara bahasa pemograman java dengan bahasa kode HTML sehingga disebut bahasa hybrid. Walaupun javascript merupakan turunan dari java namun javascript tidak memiliki aturan yang serumit java.

- Terbuka

Javascript tidak terikat oleh hardware maupun software tertentu bahkan system operasi seperti windows maupun unix. Karena ia bersifat terbuka, maka ia dapat dibuat maupun dibaca di semua jenis komputer.

) Kekurangan dari Javascript.

- Script tidak terenkripsi

Karena javascript bersifat client side, maka script yang kita buat di text editor dan telah dijadikan web di server, ketika user merequest web dari server tersebut maka sintak javascript akan langsung ditampilkan di browser. User bisa melihat dan menirunya dari sourcenya.

- **Kemampuan terbatas**

Walaupun javascript mampu membuat bentuk web menjadi interaktif dan dinamis, namun javascript tidak mampu membuat program aplikasi sendiri seperti java.

- **Keterbatasan Objek**

Javascript tidak mampu membuat kelas yang bisa menampung objek - objek tambahan seperti java karena javascript telah memiliki objek yang builtin pada struktur bahasanya.

4. Dasar – dasar Pemrograman Javascript.

Programmer menggunakan variabel untuk menyimpan nilai, sebuah variabel dapat menampung beberapa jenis data. Dalam Javascript kita tidak perlu mendeklarasikan tipe data suatu variabel sebelum menggunakannya. Setiap variabel dapat menampung semua tipe data Javascript, termasuk :

1. String Data
2. Jumlah
3. Boolean nilai (T / F)

Nama variabel ada aturan dan konvensi dalam pemrograman apapun bahasa. Ini adalah praktik yang baik untuk menggunakan nama deskriptif untuk variabel.

5. Aturan – aturan dalam Javascript.

1. Nama atau variabel harus dimulai dengan huruf atau garis bawah, contoh (kelompokDelapan atau _grupDelapan)
2. Kita dapat menggunakan nomor dalam nama variabel, tetapi bukan sebagai yang pertama nama karakter. contoh (nama15 atau biaya\$)
3. Kita tidak dapat menggunakan ruang untuk karakter terpisah.
4. Hindari gunakan huruf pertama setiap kata kecuali yang pertama. Contoh (salesTax atau userFirstNameVariabel)
5. Untuk mendeklarasikan variabel, gunakan kata kunci var dan nama variabel.
6. Untuk memberikan nilai pada variabel tambahkan tanda yang sama dan memiliki nilai.

) **Pengertian CSS**

CSS adalah singkatan dari Cascading Style Sheets. Berisi rangkaian instruksi yang menentukan bagaimana suatu text akan tertampil di halaman web.

Perancangan desain text dapat dilakukan dengan mendefinisikan fonts (huruf) , colors (warna), margins (ukuran), latar belakang (background), ukuran font (font sizes) dan lain-lain. Elemen-elemen seperti colors (warna) , fonts (huruf), sizes (ukuran) dan spacing (jarak) disebut juga “styles”. Cascading Style Sheets juga bisa berarti meletakkan styles yang berbeda pada layers (lapisan) yang berbeda.

CSS terdiri dari style sheet yang memberitahukan browser bagaimana suatu dokumen akan disajikan. Fitur-fitur baru pada halaman web lama dapat ditambahkan dengan bantuan style sheet. Saat menggunakan CSS, Anda tidak perlu menulis font, color atau size pada setiap paragraf, atau pada setiap dokumen. Setelah Anda membuat sebuah style sheet, Anda dapat menyimpan kode tersebut sekali saja dan dapat kembali menggunakannya bila diperlukan.

) **Keuntungan Menggunakan CSS**

CSS memberikan keseragaman pada halaman web. Dengan CSS dapat menghemat banyak waktu dan pekerjaan berulang. Saat menggunakan CSS, perubahan tidak perlu dilakukan dalam setiap halaman web. Anda hanya perlu membuat perubahan dalam style sheet. CSS memungkinkan Anda untuk memuat halaman web Anda dengan mudah. Layers (Lapisan), seperti item pop-up, dapat digunakan dalam dokumen. CSS membantu Anda memelihara halaman web Anda dengan mudah dan efektif.

) **TAG DIV**

Div biasanya digunakan untuk membuat layer yang akan memudahkan user untuk membuat layout sesuai dengan design yang user inginkan. Dimana untuk membuat layer user dapat menggunakan tag Div dan diberi atribut ID ataupun Class. Hal yang perlu diperhatikan dalam menggunakan tag div adalah tidak diperkenankan ada dua elemen yang memiliki ID/class dengan nama yang sama. Karena salah penggunaan ID, maka akan berdampak pada hasil tampilan web itu sendiri.

BAB II PEMBAHASAN

1. MODIFIKASI GAME PIG 1

Pemain kehilangan skor SELURUHNYA saat dia mengocok dan mendapatkan 6 dua kali. Setelah itu, giliran pemain berikutnya. (Petunjuk: Selalu simpan nilai hasil dadu kocokan sebelumnya dalam variabel terpisah)

PEMECAHAN KASUS

) Membuat coding .html

```
<!DOCTYPE html>
<html lang = "en" >
< head >
  < meta charset="UTF-8">
  <title>Pig Game By Anggi Anggraini 201501045</title>
  <link href = "https://fonts.googleapis.com/css?family=Lato:100,300,600"
rel="stylesheet" type="text/css">
  <link rel = 'stylesheet prefetch'
href='https://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css'>
  <link rel = "stylesheet" href="1style.css">
</head>
<body>
  <div class="wrapper clearfix" >
    <div class="player-0-panel active-0">
      <div class="game-panel">
        <div class="player-name" id="name-0">Player
1</div>
        <div class="player-score-0" id="score-0">43</div>
        <div class="player-0-rolled-1"><h3>Nilai dadu
1.</h3></div>
        <div class="player-current-box-0">
          <div class="player-current-
label">Current</div>
          <div class="player-current-score"
id="current-0">11</div>
        </div>
      </div>
      <div class="rules-panel" style="display:none">
        <h3>Winner</h3>
        <p>Player yang pertama mendapatkan nilai 100</p>
      </div>
    </div>
    <div class="player-1-panel">
      <div class="game-panel">
        <div class="player-name" id="name-1">Player
2</div>
        <div class="player-score-1" id="score-1">72</div>
        <div class="player-1-rolled-1"><h3>Nilai dadu
1.</h3></div>
        <div class="player-current-box-1">
          <div class="player-current-
label">Current</div>
```

```

                                <div class="player-current-score"
id="current-1">0</div>
                                </div>
                                </div>
                                <div class="rules-panel" style="display:none">
                                    <h3>Aturan Permainan</h3>
                                    <p>Silahkan Mainkan dadu anda</p>
                                    <li>Jika NILai Dadu Anda 1 maka Giliran Anda
Habis</li>
                                    <li>Jika NILai Dadu Anda = 6 secara berturut-turut selama 2 kali
maka Nilai anda jadi 0</li>
                                    <button class="btn-back"><i class="ion-ios-arrow-left
color-0"></i>Kembali Ke Permainan</button>
                                    </div>
                                </div>
                                <div class="game-panel">
                                    <button class="btn-new"><i class="ion-ios-plus-outline icon color-
0"></i>New Game</button>
                                    <button class="btn-rules"><i class="ion-ios-compose-outline
icon color-0"></i>Help</button>
                                    <button class="btn-roll"><i class="ion-ios-loop icon color-
0"></i>Mainkan</button>
                                    <button class="btn-hold"><i class="ion-ios-download-outline icon color-
0"></i>Save, and Hold</button>
                                    <img src = "1-dice-5.png" alt= "Dice" class="dice">
                                </div>
                                </div>
                                <script src = "nomer1.js" ></ script >
</ body >
</ html >

```

) Membuat coding.css

```

* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

.clearfix::after {
    content: "";
    display: table;
    clear: both;
}

body {
    background-image: linear-gradient(rgba(62, 20, 20, 0.4), rgba(62, 20, 20, 0.4)),
url(1.jpg);
    font-family: Lato;
    font-weight: 300;
    position: relative;
    height: 100vh;
    color: #555;
}

h3, p, li {text-transform: uppercase;}

.wrapper {
    width: 1000px;
    position: absolute;

```

```

    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    background-color: #fff;
    box-shadow: 0px 10px 30px rgba(0, 0, 0, 0.3);
    overflow: hidden;
}

.rules-panel {
    line-height: 30px;
}

.player-0-panel,
.player-1-panel {
    width: 50%;
    float: left;
    height: 600px;
    padding: 100px;
}

/***** PLAYERS *****/

.player-name {
    font-size: 40px;
    text-align: center;
    text-transform: uppercase;
    letter-spacing: 2px;
    font-weight: 100;
    margin-top: 20px;
    margin-bottom: 10px;
    position: relative;
}

.player-score-0 {
    text-align: center;
    font-size: 80px;
    font-weight: 100;
    color: #EB4D4D;
    margin-bottom: 130px;
}

.player-score-1 {
    text-align: center;
    font-size: 80px;
    font-weight: 100;
    color: #FF1493;
    margin-bottom: 130px;
}

.player-0-rolled-1 {
    text-align: center;
    margin-bottom: 10px;
}

.player-1-rolled-1 {
    text-align: center;
    margin-bottom: 10px;
}

```



```

}

.active-0 { background-color: rgba(255, 175, 189, 0.1) }
.active-1 { background-color: rgba(46, 204, 113, 0.1) }
.active-0 .player-name, .active-1 .player-name { font-weight: 300; }

.active .player-name::after {
  content: "\2022";
  font-size: 47px;
  position: absolute;
  color: #EB4D4D;
  top: -7px;
  right: 10px;
}

}

.player-current-box-0 {
  background-color: #EB4D4D;
  color: #fff;
  width: 40%;
  margin: 0 auto;
  padding: 12px;
  text-align: center;
}

}

.player-current-box-1 {
  background-color: #2ecc71;
  color: #fff;
  width: 40%;
  margin: 0 auto;
  padding: 12px;
  text-align: center;
}

}

.player-current-label {
  text-transform: uppercase;
  margin-bottom: 10px;
  font-size: 12px;
  color: #222;
}

}

.player-current-score {
  font-size: 30px;
}

}

button {
  position: absolute;
  width: 200px;
  left: 50%;
  transform: translateX(-50%);
  color: #555;
  background: none;
  border: none;
  font-family: Lato;
  font-size: 20px;
  text-transform: uppercase;
  cursor: pointer;
  font-weight: 300;
  transition: background-color 0.3s, color 0.3s;
}

}

button:hover { font-weight: 600; }

```

```

button:hover i { margin-right: 20px; }

button:focus {
    outline: none;
}

i {
    display: inline-block;
    margin-right: 15px;
    font-size: 32px;
    line-height: 1;
    vertical-align: text-top;
    margin-top: -4px;
    transition: margin 0.3s;
}

.color-0 {color: #EB4D4D;}
.color-1 {color: #2ecc71;}

.btn-new { top: 105px;}
.btn-rules { top: 45px;}
.btn-roll { top: 413px;}
.btn-back { top: 467px;}
.btn-hold { top: 477px;}

.dice {
    position: absolute;
    left: 50%;
    top: 178px;
    transform: translateX(-50%);
    height: 100px;
    box-shadow: 0px 10px 60px rgba(0, 0, 0, 0.10);
}

.winner-0 { background-color: #f7f7f7; }
.winner .player-name { font-weight: 300; color: #90EE90; }

.winner-0 { background-color: rgba(255, 175, 189, 0.1) }
.winner-1 { background-color: rgba(46, 204, 113, 0.1) }
.winner-0 .player-name { font-weight: 300; color: #EB4D4D;}
) .winner-1 .player-name { font-weight: 300; color: #2ecc71;}

```

) Membuat coding.js

```

var scores, roundScore, activePlayer, gamePlaying;
var btnRoll = document.querySelector('.btn-roll');
var btnHold = document.querySelector('.btn-hold');
//dice images
var diceimgs = {
    diceimg01: "dice-6.png",
    diceimg02: "dice-5.png",
    diceimg03: "dice-4.png",
    diceimg04: "dice-3.png",
    diceimg05: "dice-2.png",
    diceimg06: "dice-1.png",
    diceimg11: "dice-7.png",
    diceimg12: "dice-8.png",
    diceimg13: "dice-9.png",
    diceimg14: "dice-10.png",
    diceimg15: "dice-11.png",

```

```

diceimg16: "dice-12.png"

});
init();
var simpan;
document.querySelector('.btn-roll').addEventListener('click', function(){
    if (gamePlaying)
    {
        var dice = Math.floor(Math.random() * 6) + 1;
        var diceDOM = document.querySelector('.dice');
        diceDOM.style.display = 'block';
        diceDOM.src = diceimgs['diceimg' + activePlayer + dice];
        document.querySelector('#current-' + activePlayer).innerHTML = '<em>' + dice
+ '</em>';
        if (dice !== 1)
        {
            hideRolledMsg();
            if (dice == 6 && simpan == 6)
            {
                disableBtn(btnRoll, 1000);
                scores[activePlayer] = 0;
                document.querySelector('#score-' + activePlayer).textContent =
scores[activePlayer];
                document.querySelector('.player-' + activePlayer + '-rolled-
1').style.visibility = 'visible';
                nextPlayer();
                console.log(dice + simpan);
            }
            hideRolledMsg();
            console.log(dice + ' dan ' + simpan);
            simpan = dice;
            roundScore += dice;
            document.querySelector('#current-' + activePlayer).textContent =
roundScore;
        }
        else
        {
            disableBtn(btnRoll, 1000);
            hideRolledMsg();
            document.querySelector('.player-' + activePlayer + '-rolled-
1').style.visibility = 'visible';
            nextPlayer();
        }
    }
});

document.querySelector('.btn-hold').addEventListener('click', function(){
    if (gamePlaying)
    {
        disableBtn(btnRoll, 1000);
        scores[activePlayer] += roundScore;
        document.querySelector('#score-' + activePlayer).textContent =
scores[activePlayer];
        if (scores[activePlayer] >= 100)
        {
            document.querySelector('#name-' + activePlayer).textContent = 'Winner!';
            document.querySelector('.dice').style.display = 'none';
            document.querySelector('.player-' + activePlayer + '-
panel').classList.add('winner-' + activePlayer);

```

```

        document.querySelector('.player-' + activePlayer + '-
panel').classList.remove('active-' + activePlayer);
        gamePlaying = false;
    }
    else
    {
        nextPlayer();
    }
}
});

document.querySelector('.btn-new').addEventListener('click', init);

document.querySelector('.btn-rules').addEventListener('click', function(){
    var games = document.getElementsByClassName('game-panel');
    for (i = 0; i < games.length; i++)
    {
        games[i].style.display = 'none';
    }
    document.querySelector('.btn-back').style.display = 'block';
    var rules = document.getElementsByClassName('rules-panel');
    for (i = 0; i < rules.length; i++)
    {
        rules[i].style.display = 'block';
    }
});

document.querySelector('.btn-back').addEventListener('click', function(){
    var games = document.getElementsByClassName('game-panel');
    for (i = 0; i < games.length; i++)
    {
        games[i].style.display = 'block';
    }

    document.querySelector('.btn-back').style.display = 'none';
    var rules = document.getElementsByClassName('rules-panel');
    for (i = 0; i < rules.length; i++)
    {
        rules[i].style.display = 'none';
    }
});

function init()
{
    scores = [0, 0];
    roundScore = 0;
    activePlayer = 0;
    gamePlaying = true;
    document.querySelector('.dice').style.display = 'none';
    document.getElementById('score-0').textContent = '0';
    document.getElementById('score-1').textContent = '0';
    document.getElementById('current-0').textContent = '0';
    document.getElementById('current-1').textContent = '0';
    document.querySelector('.player-0-rolled-1').style.visibility = 'hidden';
    document.querySelector('.player-1-rolled-1').style.visibility = 'hidden';

    document.querySelector('#name-0').textContent = 'Pemain 1';
    document.querySelector('#name-1').textContent = 'Pemain 2';
    document.querySelector('.player-0-panel').classList.add('active-0');
    document.querySelector('.player-0-panel').classList.remove('winner-0');
    document.querySelector('.player-1-panel').classList.remove('winner-1');

```

```

}

function nextPlayer()
{
    //next player
    var icons = document.getElementsByTagName('i');
    for (i = 0; i < icons.length; i++)
    {
        icons[i].classList.remove('color-' + activePlayer);
    }

    document.querySelector('.dice').style.display = 'none';
    document.querySelector('.player-' + activePlayer + '-panel').classList.remove('active-' + activePlayer);
    activePlayer === 0 ? activePlayer = 1 : activePlayer = 0;
    roundScore = 0;

    for (i = 0; i < icons.length; i++)
    {
        icons[i].classList.add('color-' + activePlayer);
    }
    document.querySelector('.player-' + activePlayer + '-panel').classList.add('active-' + activePlayer);
    document.querySelector('#current-0').textContent = '0';
    document.querySelector('#current-1').textContent = '0';
}

function disableBtn(btn, time)
{
    //disable button
    btn.disabled = true;
    setTimeout(function(){ btn.disabled = false; },time);
}

function hideRolledMsg()
{
    document.querySelector('.player-0-rolled-1').style.visibility = 'hidden';
    document.querySelector('.player-1-rolled-1').style.visibility = 'hidden';
}

```

) Analisis

Game diatas merupakan Pig Game.Dimana terdiri dari dua player dan 1 dadu yang dimainkan dimana Pemain/Player akan kehilangan skor seluruhnya apabila saat ia mengkocok dadu mendapatkan nilai mata dadu berjumlah 6 angka sebanyak dua kali.Dan ketika terjadi kondisi tersebut maka giliran player selnjutnya yang akan bermain.

Langkah dalam membuat gambe ini cukup sederhana,yaitu dengan cara membuat coding html terlebih dahulu dilanjutkan membuat coding.css nya dimana coding.css berguna untuk menyesuaikan posisi tampilannya/UI.Dan yang terakhir adalah membuat logika dalam javascripnya.

) Output



2. MODIFIKASI GAME PIG 2

Tambahkan bidang input ke HTML di mana pemain dapat mengatur skor kemenangan, sehingga mereka dapat mengubah skor standar 100. (Petunjuk: Anda dapat membaca nilai itu dengan properti **.value** di JavaScript gunakan google untuk mencari tahu ini)

) Membuat coding .html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Pig Game By Anggi Anggraini 201501045</title>
    <link href = "2style.css" rel="stylesheet" type="text/css"/>
  </head>
  <body>
    <center>
      <div class="start-menu container">
        <div class="what">
          <h1><i>Welcome in Pig Game, Please input Your Name</i></h1>
          <form id = "player-1" >
            <div class="form-group">
              <label for="player1Name">Player 1:</label><br>
              <input required class="player1Name" type="text" placeholder="Input name
of player 1">
            </div>
          </form>
          <form id = "player-2" >
            <div class="form-group">
              <label for="player2Name">Player 2:</label><br>
              <input required class="player2Name" type="text" placeholder="Input name
of player 2">
            </div>
          </form>
          <button id = "start" class="btn">Start</button>
        </div>
      </div>
    </center>
    <div class="player-console container">
      <form action = "2modul05.html" >
        <input type="submit" class="btn" value="New Game" />
      </form>
      <div class = "row">
        <div class="col-md-6">
          <h2>Player 1: <span id = "player1Name" ><span ></ h2 >
          <button id="player1-roll" class="btn">Kocok</button>
          <button id = "player1-hold" class="btn">Ambil</button>
          <h2 id = "die-roll-1" > Nilai Dadu : </h2>
          <p id = "round-total-1" > Total Ronde : </p>
          <p id = "total-score-1" > Skor Player 1 : </p>
        </div>

        <div class="col-md-6">
          <h2>Player 2: <span id = "player2Name" ><span ></ h2 >
          <button id="player2-roll" class="btn">Kocok</button>
          <button id = "player2-hold" class="btn">Ambil</button>
          <h2 id = "die-roll-2" > Nilai Dadu : </h2>
          <p id = "round-total-2" > Total Ronde : </p>
          <p id = "total-score-2" > Skor Player 2 : </p>
        </div>
      </div>
    </div>
  </body>
</html>
```

```

        <img src = "dice-5.png" align="center" alt="Dice" class="dice">
    </div>
</div>
</div>
<script src = "min.js" ></ script >
< script src="nomer2.js"></script>
</body>
</html>

```

) Membuat coding .css

```

body{
    font-family: 'Hind Siliguri', Open Sans;
    background-color: #FFFAF0;
}
.player-console {
    display: none;
}
.what{
    height: 450px;
}
input[type=text], select {
    padding: 10px 16px;
    margin: 8px 0;
    display: inline-block;
    border: 2px solid #ccc;
    box-sizing: border-box;
    background-color: #ccc;
}
.btn{
    background-color: #FF1493;
    color: white;
    font-size: 18px;
    padding: 12px 20px;
    margin: 8px 0;
    border: none;
    border-radius: 4px;
    cursor: pointer;
    margin-left: 30px;
}
.dice {
    position: absolute;
    left: 50%;
    top: 178px;
    transform: translateX(-50%);
    height: 100px;
    box-shadow: 0px 10px 60px rgba(0, 0, 0, 0.10);
}

```


) Membuat coding .js

```
//business logic
var player1 = "";
var player2 = "";

var throwdice = function() {
    return Math.floor(6 * Math.random()) + 1;
}

function Player(turn)
{
    this.roll = 0;
    this.tempscore = 0;
    this.totalscore = 0;
    this.turn = turn;
    this.playerName;
}

Player.prototype.rollone = function()
{
    if (this.roll === 1)
    {
        this.tempscore = 0;
        alert("Maaf" + this.playerName + ", Nilai dadu kamu adalah 1. Permainanmu telah selesai kini giliran lawan yang bermain.");
    }
    else
    {
        this.tempscore += this.roll;
    }
}

Player.prototype.hold = function()
{
    this.totalscore += this.tempscore;
    this.tempscore = 0;
    alert(this.playerName + ", giliranmu sudah selesai, sekarang gerakan mousenya");
}

Player.prototype.winnerCheck = function()
{
    if (this.totalscore >= 100)
    {
        alert(this.playerName + " The Winner");
    }
}

Player.prototype.newGame = function()
{
    this.roll = 0;
    this.tempscore = 0;
    this.totalscore = 0;
    this.playerName = "";
}

// User Interface
$(document).ready(function() {

    $("button#start").click(function(event) {
        player1 = new Player(true);
        player2 = new Player(false);
    });
});
```

```

$(".player-console").show();
$(".start-menu").hide();

    var player1Name = $(".player1Name").val();
    $("#player1Name").text(player1Name);

    var player2Name = $(".player2Name").val();
    $("#player2Name").text(player2Name);

    player1.playerName = player1Name;
    player2.playerName = player2Name;

});

$("#button#player1-roll").click(function(event) {
    player1.roll = throwdice();
    $("#die-roll-1").text(player1.roll);
    player1.rollone();
    document.querySelector('.dice').src = 'gambar/dice-' + player1.roll +
'.png';
    $("#round-total-1").text(player1.tempscore);
});

$("#button#player2-roll").click(function(event) {
    player2.roll = throwdice();
    $("#die-roll-2").text(player2.roll);
    player2.rollone();
    document.querySelector('.dice').src = 'gambar/dice-' + player2.roll +
'.png';
    $("#round-total-2").text(player2.tempscore);
});

$("#button#player1-hold").click(function(event) {
    player1.hold();
    $("#total-score-1").text(player1.totalscore);
    $("#round-total-1").empty();
    $("#die-roll-1").empty();
    player1.winnerCheck();
});

$("#button#player2-hold").click(function(event) {
    player2.hold();
    $("#total-score-2").text(player2.totalscore);
    $("#round-total-2").empty();
    $("#die-roll-2").empty();
    player2.winnerCheck();
});
});

```

) Membuat coding min.js

```

1 i.length>2&&"ID"===j[i[e]].type&&9===b.nodeType&&p&&d.relative[i[1].type]){if(b=(d.find.ID(j.matches[0].replace(_aa,b)))[0],b)return c;m&&(b
2 ght","width"]},function(a,b){r.cssHooks[b]={get:function(a,u){if(!r)return Qa.test(r.css(a,"display"))?a.getClientRects().length&&a.getBoundingCl
3 d,r});var Vb=a.jquery,Wb=a.$;return r.noConflict=function(b){return a.$=r&&(a.$=Wb),b&&a.jquery=r&&(a.jquery=Wb),r},b)&&(a.jquery=a.$=r),r)};
4

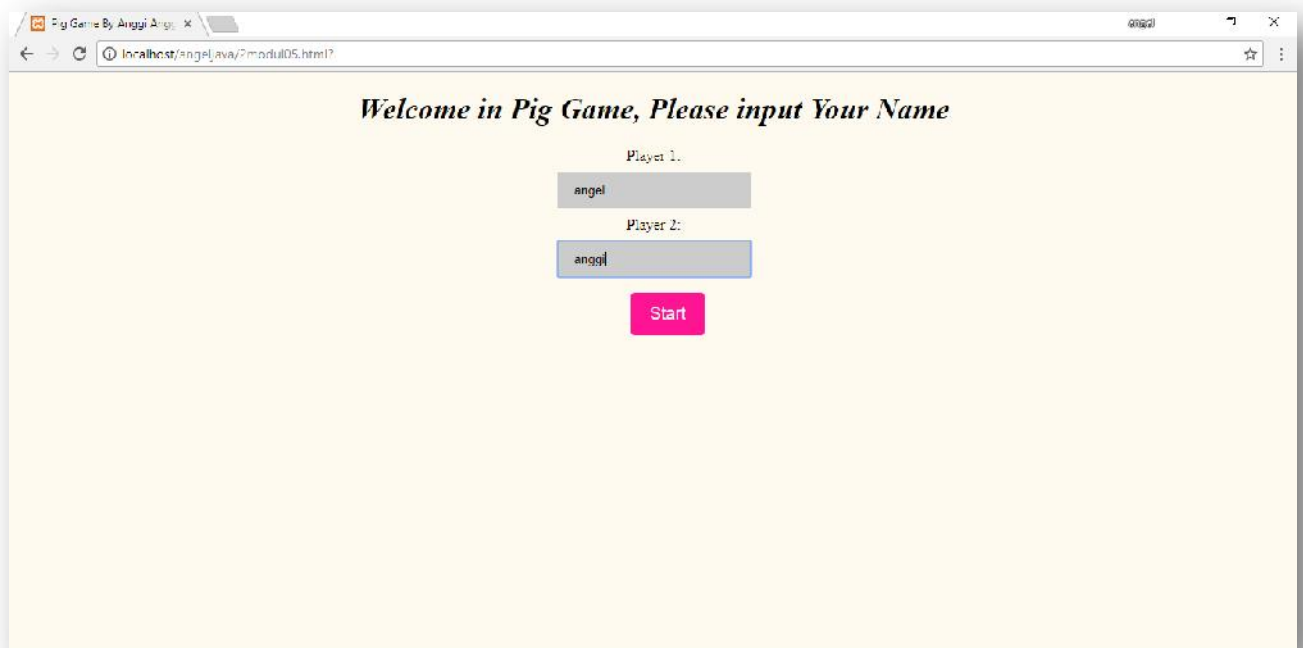
```

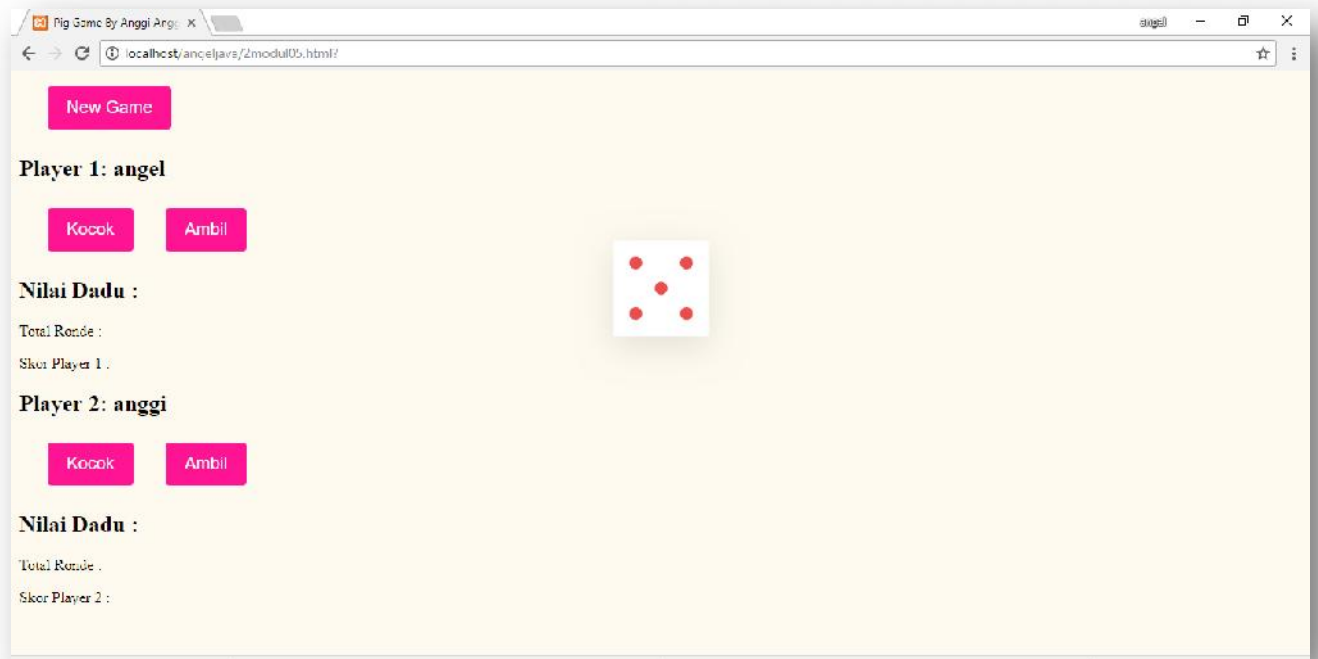
) Analisis

Game diatas merupakan Pig Game.Dimana pemain dapat mengatur skor kemenangannya masing-masing. Sehingga mereka dapat mengubah skor standar yang telah ditentukan sebelumnya oleh game tersebut dengan skor yang telah ditentukan oleh pemain itu sendiri.

Langkah dalam membuat game ini cukup sederhana, yaitu dengan cara membuat coding html terlebih dahulu dilanjutkan membuat coding.css nya dimana coding.css berguna untuk menyesuaikan posisi tampilannya/UI,membuat logika dalam javascriptnya.dan membuat 1 query min.js.

) Output





3. MODIFIKASI GAME PIG 3

Tambahkan dadu lain ke permainan, sehingga sekarang ada dua dadu. Pemain kehilangan skor saat ini ketika salah satunya adalah 1. (Petunjuk: Anda akan membutuhkan CSS untuk memposisikan dadu kedua, karena itu lihatlah kode CSS.)

) Membuat Coding .html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Pig Game By Anggi Anggraini 201501045</title>
    <link href = "3style.css" rel="stylesheet" type="text/css"/>
  </head>
  <body>
    <center>
      <div class="start-menu container">
        <div class="what">
          <h1><i>Welcome in Pig Game, Please input Your Name</i></h1>
          <form id = "player-1" >
            <div class="form-group">
              <label for="player1Name">Player 1:</label><br>
              <input required class="player1Name" type="text" placeholder="Input name
of player 1">
            </div>
          </form>
          <form id = "player-2" >
            <div class="form-group">
              <label for="player2Name">Player 2:</label><br>
              <input required class="player2Name" type="text" placeholder="Input name
of player 2">
            </div>
          </form>
          <button id = "start" class="btn">Start</button>
        </div>
      </div>
    </center>
    <div class="player-console container">
      <form action = "3modul05.html" >
        <input type="submit" class="btn" value="New Game" />
      </form>
      <div class = "row">
        <div class="col-md-6">
          <h2>Player 1: <span id = "player1Name" ><span ></ h2 >
            <button id="player1-roll" class="btn">Kocok</button>
            <button id = "player1-hold" class="btn">Ambil</button>
            <p><h2 id = "die-roll-1" > Nilai Dadu 1 : </h2></p>
            <p><h2 id = "die-roll-1-1" > Nilai Dadu 2 : </h2></p>
            <p id = "round-total-1" > Total Ronde : </p>
            <p id = "total-score-1" > Skor Player 1 : </p>
          </div>
          <div class="col-md-6">
            <h2>Player 2: <span id = "player2Name" ><span ></ h2 >
              <button id="player2-roll" class="btn">Kocok</button>
              <button id = "player2-hold" class="btn">Ambil</button>
              <p><h2 id = "die-roll-2" > Nilai Dadu 1 : </h2></p>
              <p><h2 id = "die-roll-2-2" > Nilai Dadu 2 : </h2></p>
              <p id = "round-total-2" > Total Ronde : </p>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

```

        <p id = "total-score-2" > Skor Player 2 : </p>
        <p><img src = "gambar/dice-5.png" align="center" alt="Dice"
class="dice"></p>
        <img src = "gambar/dice-5.png" align="right" alt="Dice2" class="dice2">
        </div>
        </div>
        </div>
<script src = "mine.js" ></ script >
< script src="nomer3.js"></script>
</body>
</html>

```

) Membuat coding .css

```

body{
    font-family: 'Hind Siliguri', Open Sans;
    background-color: #FFFAF0;
}
.player-console {
    display: none;
}
.what{
    height: 450px;
}
input[type=text], select {
    padding: 10px 16px;
    margin: 8px 0;
    display: inline-block;
    border: 2px solid #ccc;
    box-sizing: border-box;
    background-color: #ccc;
}
.btn{
    background-color: #FF1493;
    color: white;
    font-size: 18px;
    padding: 12px 20px;
    margin: 8px 0;
    border: none;
    border-radius: 4px;
    cursor: pointer;
    margin-left: 30px;
}
.dice {
    position: absolute;
    left: 50%;
    top: 178px;
    transform: translateX(-50%);
    height: 100px;
    box-shadow: 0px 10px 60px rgba(0, 0, 0, 0.10);
}

```

) Membuat coding .js

```
var player1 = "";
var player2 = "";
var throwdice = function() {
    return Math.floor(6 * Math.random()) + 1;
}
function Player(turn) {
    this.roll = 0;
    this.roll2=0;
    this.tempscore = 0;
    this.totalscore = 0;
    this.turn = turn;
    this.playerName;
}

Player.prototype.rollone = function() {
    if (this.roll === 1 || this.roll2 === 1) {
        this.tempscore = 0;
        alert("Maaf " + this.playerName + ", Nilai dadu kamu adalah 1. Permainanmu telah selesai kini giliran lawan yang bermain.");
    }
    else {
        var hasil = this.roll+this.roll2;
        this.tempscore += hasil;
    }
}

Player.prototype.hold = function() {
    this.totalscore += this.tempscore;
    this.tempscore = 0;
    alert(this.playerName + ", giliranmu sudah selesai, sekarang gerakan mousenya");
}

Player.prototype.winnerCheck = function() {
    if (this.totalscore >= 100) {
        alert(this.playerName + " adalah Pemenangnya");
    }
}

Player.prototype.newGame = function() {
    this.roll = 0;
    this.roll2=0;
    this.tempscore = 0;
    this.totalscore = 0;
    this.playerName = "";
}

// User Interface
$(document).ready(function() {

    $("#button#start").click(function(event) {
        player1 = new Player(true);
        player2 = new Player(false);
        $(".player-console").show();
        $(".start-menu").hide();

        var player1Name = $(".player1Name").val();
        $("#player1Name").text(player1Name);

        var player2Name = $(".player2Name").val();
        $("#player2Name").text(player2Name);
    });
});
```

```

    player1.playerName = player1Name;
    player2.playerName = player2Name;

});

$("#button#player1-roll").click(function(event) {
    player1.roll = throwdice();
    $("#die-roll-1").text(player1.roll);
    document.querySelector('.dice').src = 'dice-' + player1.roll + '.png';
    player1.roll2 = throwdice();
    $("#die-roll-1-1").text(player1.roll2);
    document.querySelector('.dice2').src = 'dice-' + player1.roll2 + '.png';
    player1.rollone();
    $("#round-total-1").text(player1.tempscore);
});

$("#button#player2-roll").click(function(event) {
    var lanjut=player2;
    player2.roll = throwdice();
    $("#die-roll-2").text(player2.roll);
    document.querySelector('.dice').src = 'dice-' + player2.roll + '.png';
    player2.roll2 = throwdice();
    $("#die-roll-2-2").text(player2.roll2);
    document.querySelector('.dice2').src = 'dice-' + player2.roll2 + '.png';
    player2.rollone();
    $("#round-total-2").text(player2.tempscore);
});

$("#button#player1-hold").click(function(event) {
    player1.hold();
    $("#total-score-1").text(player1.totalscore);
    $("#round-total-1").empty();
    $("#die-roll-1").empty();
    $("#die-roll-1-1").empty();
    player1.winnerCheck();
});

$("#button#player2-hold").click(function(event) {
    player2.hold();
    $("#total-score-2").text(player2.totalscore);
    $("#round-total-2").empty();
    $("#die-roll-2").empty();
    $("#die-roll-2-2").empty();
    player2.winnerCheck();
});

});

```

) Membuat coding mine.js

```

1  ,a=a.replace(/<script>/gi,'<script>');
2  ,a=a.replace(/</script>/gi,'</script>');
3  this.nodeType?var b=a(this,a);b.appendChild(a)}},prepend:function(){return .a(this,arguments,function(a){!1---this.nodeType||1---this.nodeType;
4  a){if(a.crossDomain){var b,c;return{send:function(e,f){b=r("<script>").prop({charset:a.charset,src:a.url}).on("load error",c=function(a){b.re

```

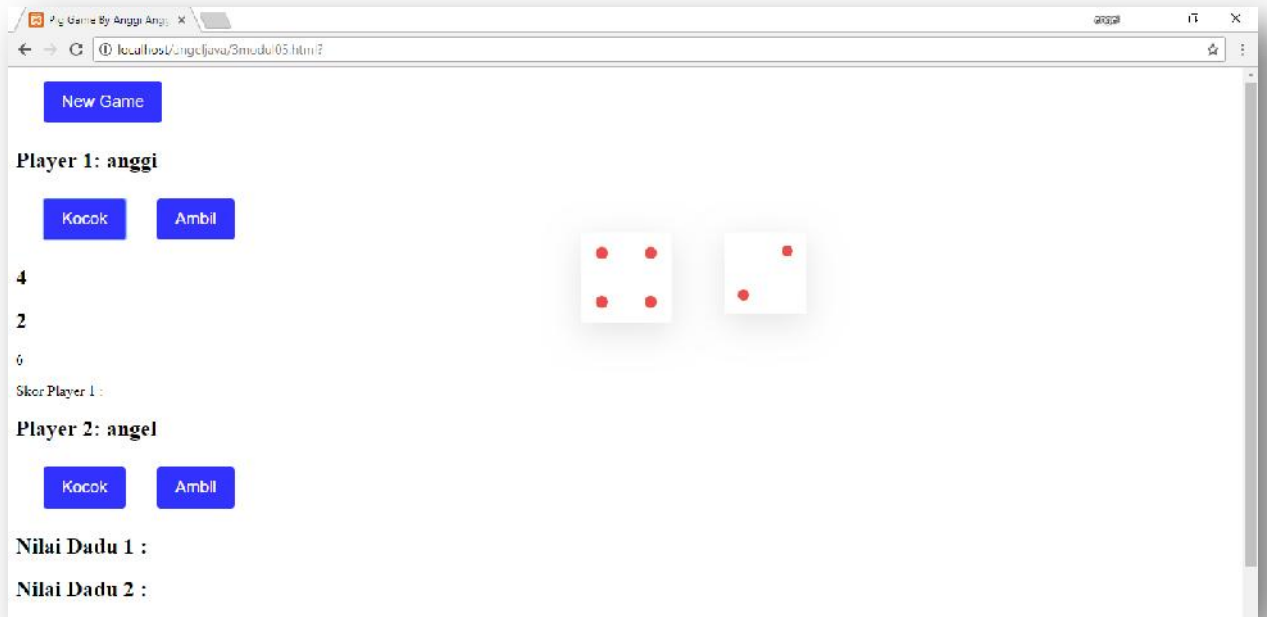

) Analisis

Game diatas merupakan Pig Game.Dimana permainan ini terdiri dari dua pemain dan dengan buah dadu yang digunakan.Dan ketika salah satu pemain mendapatkan nilai mata dadu 1 maka secara otomatis pemain tersebut akan kehilangan skornya.

Langkah dalam membuat game ini cukup sederhana,yaitu dengan cara membuat coding html terlebih dahulu dilanjutkan membuat coding.css nya dimana coding.css berguna untuk menyesuaikan posisi tampilannya/UI,membuat logika dalam javascrinya.dan membuat 1 query mine.js.

) Output





BAB III

KESIMPULAN

Dari pembahasan diatas dapat disimpulkan bahwa Pig Game merupakan sebuah permainan,dengan menggunakan media dadu. Dalam pembuatan game ini diperlukan 3 file yaitu .html, css dan .js .Dimana masing-masing file memiliki fungsi dan kegunaannya masing-masing. Seperti halnya .html dapat digunakan untuk memanggil isi dari .css dan .js sedangkan .css digunakan untuk mengatur posisi desain tampilan gamenya/UI dan yang terakhir adalah.js digunakan untuk menuliskan perintah dengan logika yang ada.termasuk logika peraturan yang diinginkan oleh si pembuat game ini,