

Laporan Pratikum Pemograman WEB Pratikum Pembuatan Game JavaScript

Disusun untuk memenuhi tugas mata kuliah Pemrograman Web yang diampu Bapak Dr. Eng.

Agussalim, M.T.



OLEH :

Nama : Achmad Yusuf Al Ma'ruf

NPM : 20082010148

Paralel : D

PROGRAM STUDI SISTEM INFOMRASI

FAKULTAS ILMU KOMPUTER

UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN"

JAWA TIMUR

2022

Tujuan Pratikum :

- Memahami konsep Java Script
- Mampu mengoperasikan struktur JavaScripts
- Memahami canvas pada HTML 5

Teori Dasar :

- **DOM**
DOM adalah sebuah platform dan interface yang memperbolehkan pengaksesan dan perubahan pada konten, struktur, dan style pada sebuah dokumen oleh program dan script
- **Canvas**
Elemen "kanvas" HTML digunakan untuk menggambar grafik melalui JavaScript. Elemen "kanvas" hanyalah wadah untuk grafik. Canvas memiliki beberapa metode untuk menggambar jalur, kotak, lingkaran, teks, dan menambahkan gambar. Canvas akan menjadi area persegi panjang pada halaman HTML. Secara default, canvas tidak memiliki batas dan tidak ada konten. Canvas pada HTML dilukis dengan menggunakan code javascript dimana Properti yang Didukung: Properti seperti Colors, Styles, Shadows, Line Styles, Rectangles, Paths, Transformations, Text, Pixel Manipulation, Compositing & Image Drawing, adalah atribut global yang didukung oleh semua tag canvas

Tools dan Software :

Tools dan software yang digunakan pada kegiatan praktikum kali ini diantaranya :

- Visual Studio Code
- Laptop
- Chrome Browser
- Internet
- Github
- Live server add on
- Asset games pada itch.io

Langkah Kerja :

1. Mempersiapkan laptop yang telah terinstal *Visual Code*
2. Install add on live server
3. Mempersiapkan bahan dan asset
4. Mendownload assets pada itch.io
5. Membuka visual code
6. Membaca dan mengcoding sesuai intruksi di modul

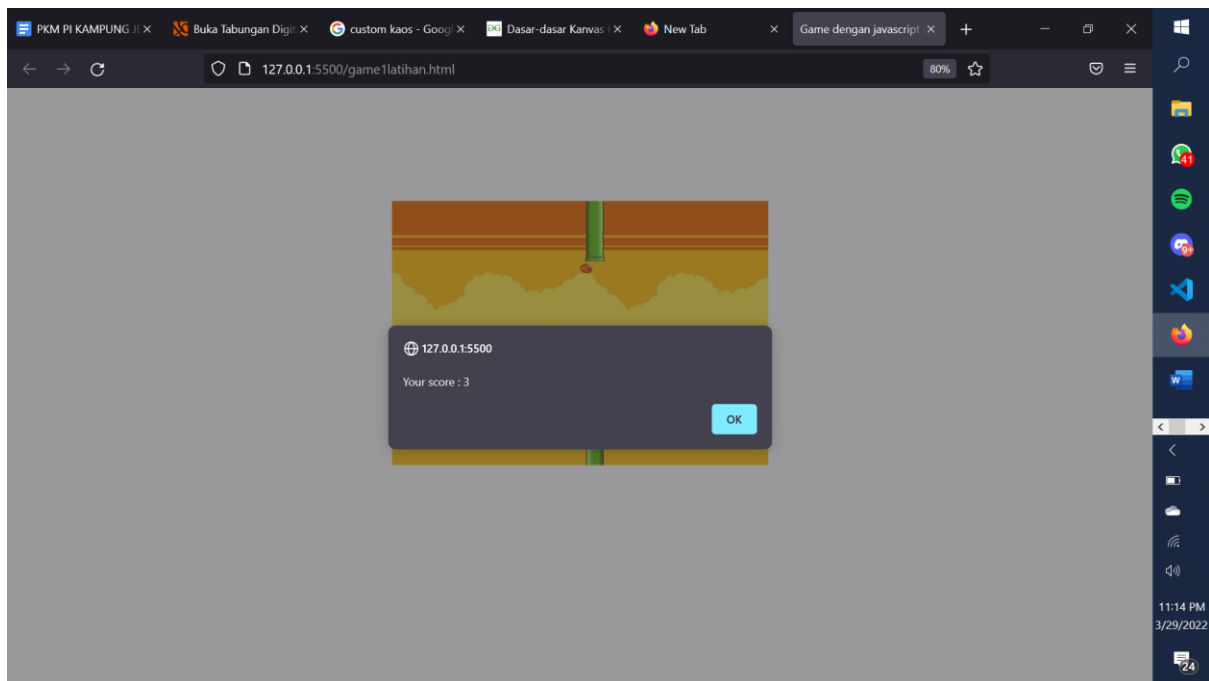
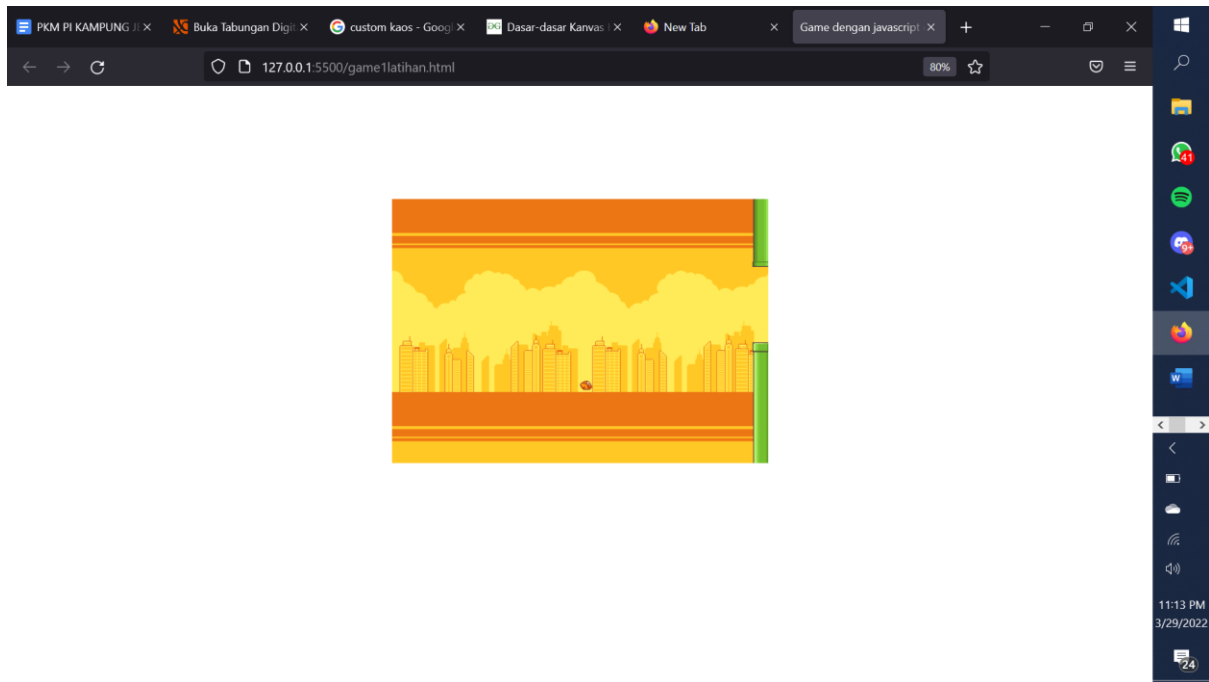
7. Menampilkan hasil kodingan dengan menekan live server
8. Jika suda selesai folder tersebut kita upluod ke github.

Link Github :

<https://github.com/yusufmaruf/gameweb.git>

GAME 1MODIFIKASI :

Tampilan :



Source code :

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8">
```

```

<title> Game dengan javascript dan HTML </title>
<style type="text/css">
    /* digunakan untuk mengatur padding dan margin untuk semua elemen */

    * {
        padding: 0;
        margin: 0;
    }
    /* Pengaturan css pada canvas */

    canvas {
        /* setting bg css pada canvas */
        background-image: url('/Background1.png');
        /* Pengaturan display css pada canvas */
        display: block;
        /* Pengaturan margin pada canvas */
        margin: 150px auto;
    }
</style>
</head>

<body>
    <!--deklarasi pada canvas-->
    <canvas id="myCanvas" width="500" height="350"></canvas>
    <script>
        //memberikan nilai pada canvas dengan seleksi dom
        var canvas = document.getElementById('myCanvas');
        //memberikan akses pada canvas untuk menggambar 2d
        var context = canvas.getContext('2d');
        //deklarasi variabel speed
        var speed = 10;
        //pemberian nilai game speed dengan 100
        gameSpeed = 100;
        //deklarasi untuk gambar burung
        var birdbg = new Image();
        //pemberian nilai gambar burung
        birdbg.src = ("/bird1.gif")
        //deklarasi variable bird dengan tipe array
        var pipe1 = new Image();
        pipe1.src = ("/pipe2.png")
        var pipe2 = new Image();
        pipe2.src = ("/pipe.png");
        var bird = {
            x: canvas.width / 2,
            y: canvas.height / 2,
            vx: 0,
            vy: 1,
        };
    </script>

```

```

//deklarasi variabel pip / tower
var pip = [];
//deklarasi grafiti
var gravity = 0.2;
//deklarasi bounce
var bounce = 4;
//deklarasi jarak
var distance = 100;
//deklarasi dir
var dir = 'left';

//penegcekan ketika menerima inputan / tombol ditekan
document.onkeydown = function(e) {
    if (!e) {
        e = window.e;
    }
    //pemberian nilai pada variabel code dengan e.keyCode
    var code = e.keyCode;
    //melakukan pengecekan apakah e.charcode dan codee = 0
    if (e.charCode && codee == 0) {
        //pemberian nilai pada code
        code = e.charCode;
    }
    switch (code) {
        //jika user menekan panah kiri
        case 37:
            if (dir != 'right')
                //maka akan memberikan nilai pada dir dengan left
                dir = 'left';
            break;
            //jika user menekan tombol panah atas
        case 38:
            //maka akan memberikan nilai pada dir dengan up

            if (dir != 'down')

                dir = 'up';
            break;
            //jika user menekan tombol panah kanan

        case 39:
            if (dir != 'left')
                //maka akan memberikan nilai pada dir dengan right
                dir = 'right';
            break;
            //jika user menekan tombol panah bawah

        case 40:

```

```

        if (dir !== 'up')
            //maka akan memberikan nilai pada dir dengan bawah
            dir = 'down';
        break;
    }

}

//fungsi membuat burung
function makeBird() {
    context.beginPath(); //untuk memulai
    context.rect(bird.x, bird.y, 10, 10); //membuat ukuran burung
    context.drawImage(birdbg, bird.x, bird.y); // membuat posisi dan
gambar burung
    context.closePath(); //mengembalikan ke nilai awal
    bird.y += bird.vy; //memberikan nilai pada bird.y dengan nilai
bird.vy ini berdampak pada posisi ketinggian burung
    bird.vy += gravity; //memberikan nilai bird.vy dengan menambahkannya
dengan nilai gravity
    if (dir === 'up') {
        bird.vy = -bounce;
        dir = '';
    }

    if (bird.y + 10 >= canvas.height) {
        // dir = 'up'
    }
}

//pembuatan halang rintang
function makePip() {
    for (p of pip) {
        context.beginPath(); //memulai path
        context.drawImage(pipe1, p.top.x, p.top.y, p.top.width = 25,
p.top.height); //pengaturan halang rintang
        context.closePath(); //pengembalian titik awal
        //setting jarak
        p.bottom.y = p.top.height + distance;
        p.bottom.height = (canvas.height - p.bottom.y);

        context.beginPath(); //memulai path
        context.drawImage(pipe2, p.bottom.x, p.bottom.y,
p.bottom.width = 25, p.bottom.height); //pengaturan halang
rintang
        context.closePath(); //pengembalian titik awal
        context.fillStyle = 'red'; //inisialisasi warna
        context.fill(); //pemberian warna
        context.closePath(); // memulai di titik awal

        p.bottom.x -= 1;
    }
}

```

```

        p.top.x -= 1;
        //penghapuan obyek di awal jika berada posisi p.top.x < -10
        if (p.top.x < -10) {
            pip.shift();
        }
        //jika obyek terkena halang rintang
        if (isCollideB(bird, p.bottom) || isCollideT(bird, p.top) ||
bird.y + 10 >= canvas.height) {
            alert('Your score : ' + score);
            window.location.reload();
        }
        //penambahan score
        if (score % 10 == 0) {
            console.log(distance);
            score += 1;
            distance -= 5;
        }
    }
}
//pengembalian nilai collision pada pipa
function isCollideB(b, p) {
    return (
        (b.x + 10 > p.x) &&
        (b.x < p.x + 10) &&
        (b.y > p.y || b.y + 10 > p.y)
    );
}
//setting collision pada bird
function isCollideT(b, p) {
    return (
        (b.x + 10 > p.x) &&
        (b.x < p.x + 10) &&
        (b.y < p.y + p.height)
    );
}
//pengaturan fungsi mtk
function rnd(min, max) {
    var r = Math.round(Math.random() * (max - min) + min);
    return (r - r % speed);
}

var score = 0;
var frame = 0;
(function loop() {
    requestAnimationFrame(function() {
        context.clearRect(0, 0, canvas.width, canvas.height);
        if (frame % gameSpeed == 0) {

```

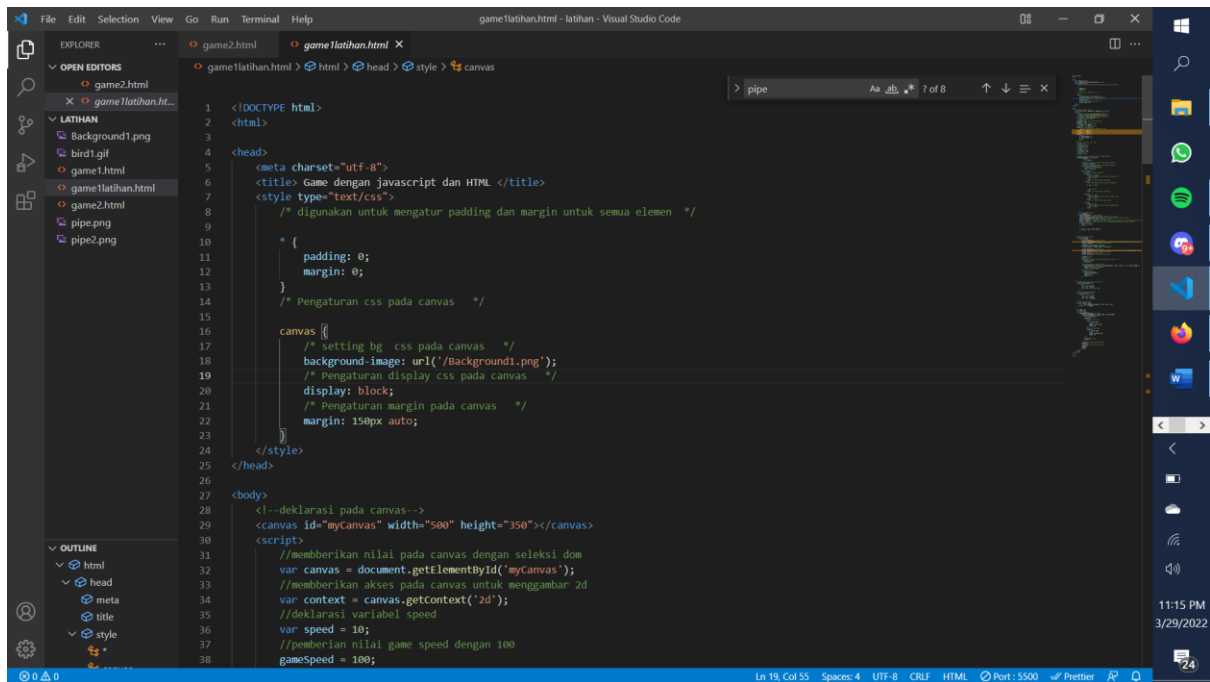


```

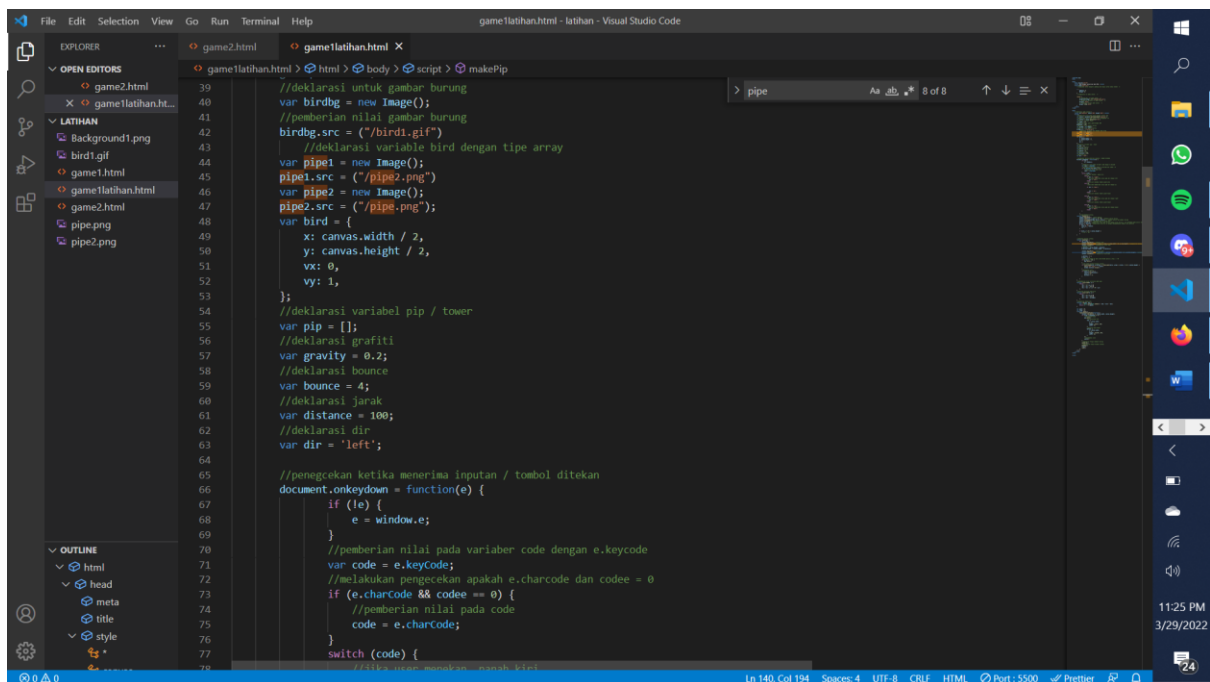
        //menampilkan pipa
        pip.push({
            //setting pipa atas
            top: {
                x: canvas.width,
                y: 0,
                height: rnd(70, 140),
                width: 10,
            },
            //setting pipa bawah
            bottom: {
                x: canvas.width,
                y: 0,
                height: rnd(100, 160),
                width: 10,
            }
        });
        //penambahan score
        score++;
    }
    //pemanggilan fungsi membuat burung
    makeBird();
    //pemanggilan fungsi halang rintang
    makePip();
    frame++
    loop();
    })
    })();
</script>
</body>

</html>

```

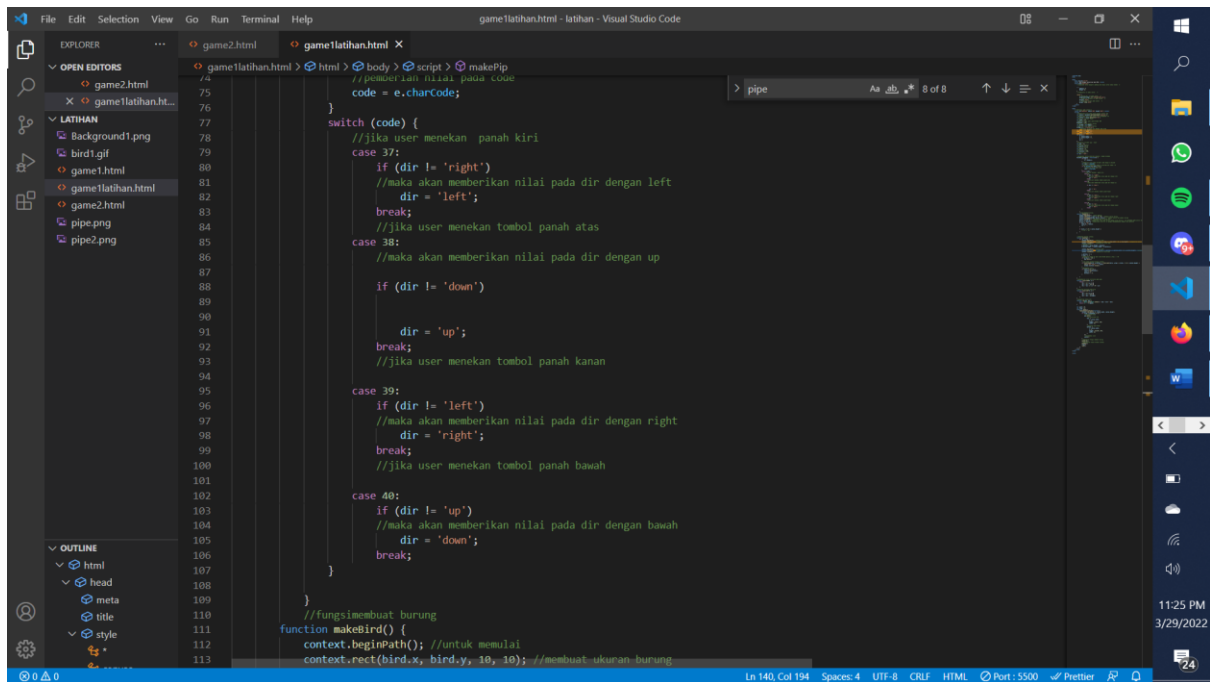


```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title> Game dengan javascript dan HTML </title>
  <style type="text/css">
    /* digunakan untuk mengatur padding dan margin untuk semua elemen */
    * {
      padding: 0;
      margin: 0;
    }
    /* Pengaturan css pada canvas */
    canvas {
      /* setting bg css pada canvas */
      background-image: url('/Background1.png');
      /* Pengaturan display css pada canvas */
      display: block;
      /* Pengaturan margin pada canvas */
      margin: 150px auto;
    }
  </style>
</head>
<body>
  <!-- deklarasi pada canvas -->
  <canvas id="myCanvas" width="500" height="350"></canvas>
  <script>
    //memberikan nilai pada canvas dengan seleksi dom
    var canvas = document.getElementById('myCanvas');
    //memberikan akses pada canvas untuk menggambar 2d
    var context = canvas.getContext('2d');
    //deklarasi variabel speed
    var speed = 10;
    //pemberian nilai game speed dengan 100
    gameSpeed = 100;
```

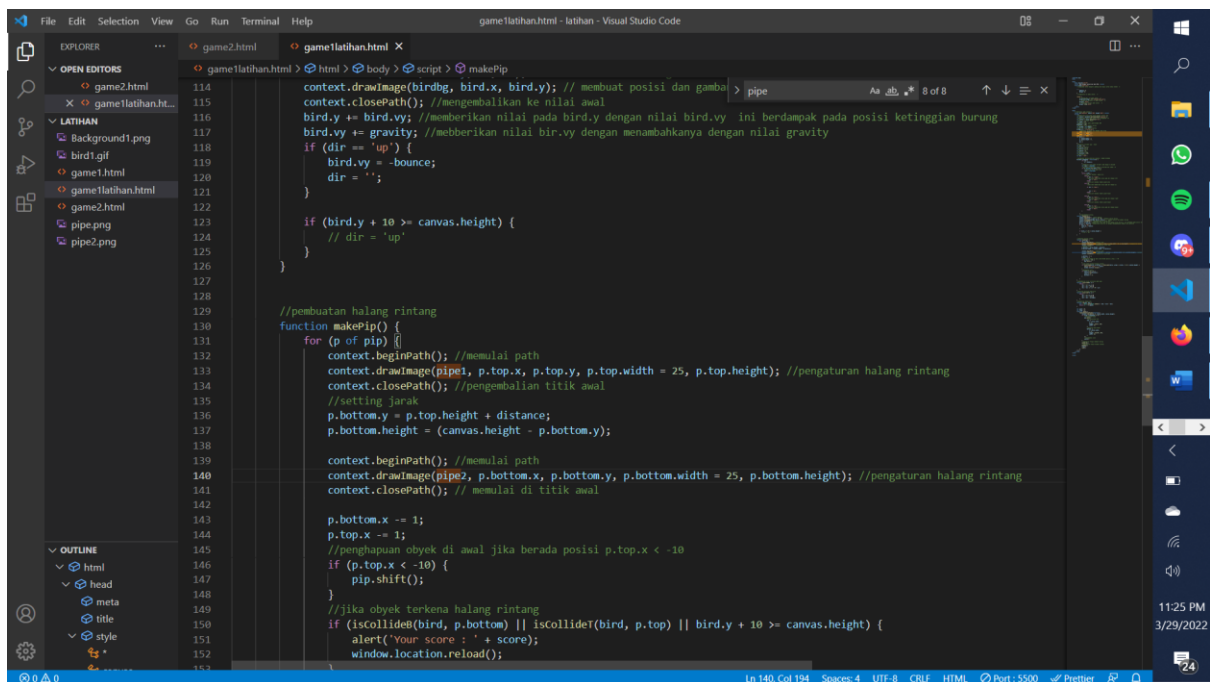


```
//deklarasi untuk gambar burung
var birdbg = new Image();
//pemberian nilai gambar burung
birdbg.src = ('/bird1.gif');
//deklarasi variabel bird dengan tipe array
var pipe1 = new Image();
pipe1.src = ('/pipe2.png');
var pipe2 = new Image();
pipe2.src = ('/pipe2.png');
var bird = {
  x: canvas.width / 2,
  y: canvas.height / 2,
  vx: 0,
  vy: 1,
};
//deklarasi variabel pip / tower
var pip = [];
//deklarasi grafiti
var gravity = 0.2;
//deklarasi bounce
var bounce = 4;
//deklarasi jarak
var distance = 100;
//deklarasi dir
var dir = 'left';

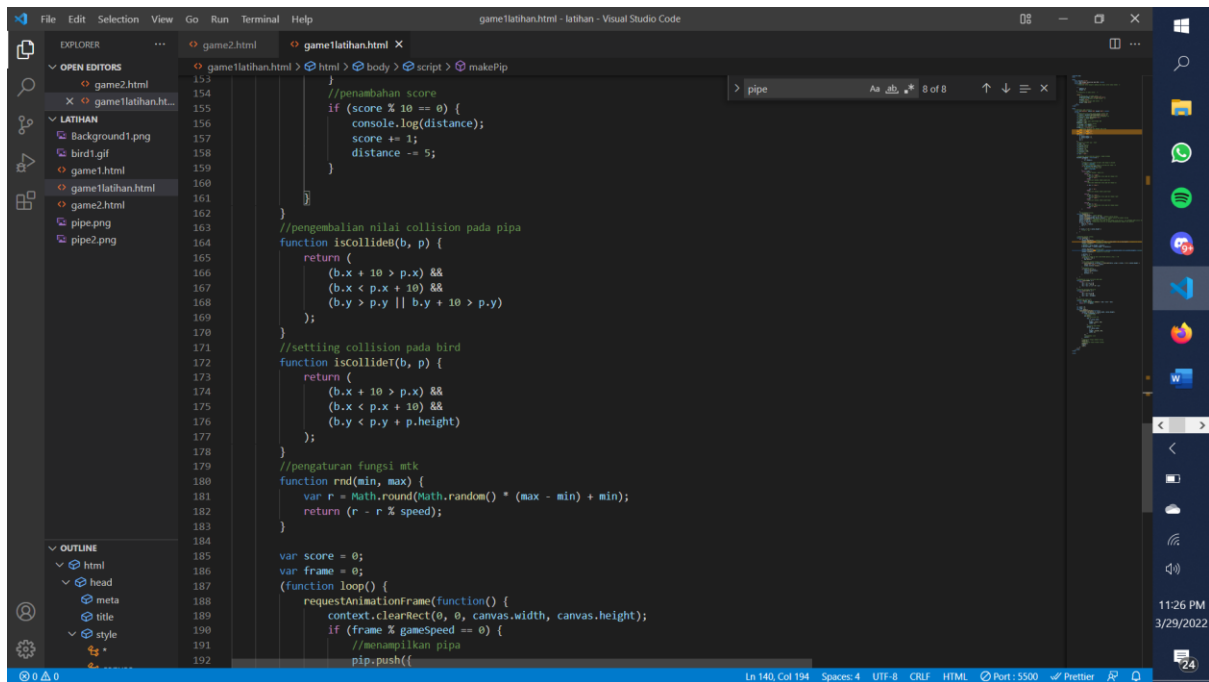
//pengecekan ketika menerima inputan / tombol ditekan
document.onkeydown = function(e) {
  if (le) {
    e = window.e;
  }
  //pemberian nilai pada variabel code dengan e.keycode
  var code = e.keyCode;
  //melakukan pengecekan apakah e.charcode dan codee = 0
  if (e.charcode && codee == 0) {
    //pemberian nilai pada code
    code = e.charcode;
  }
  switch (code) {
    //tombol untuk membuat patch bird
```



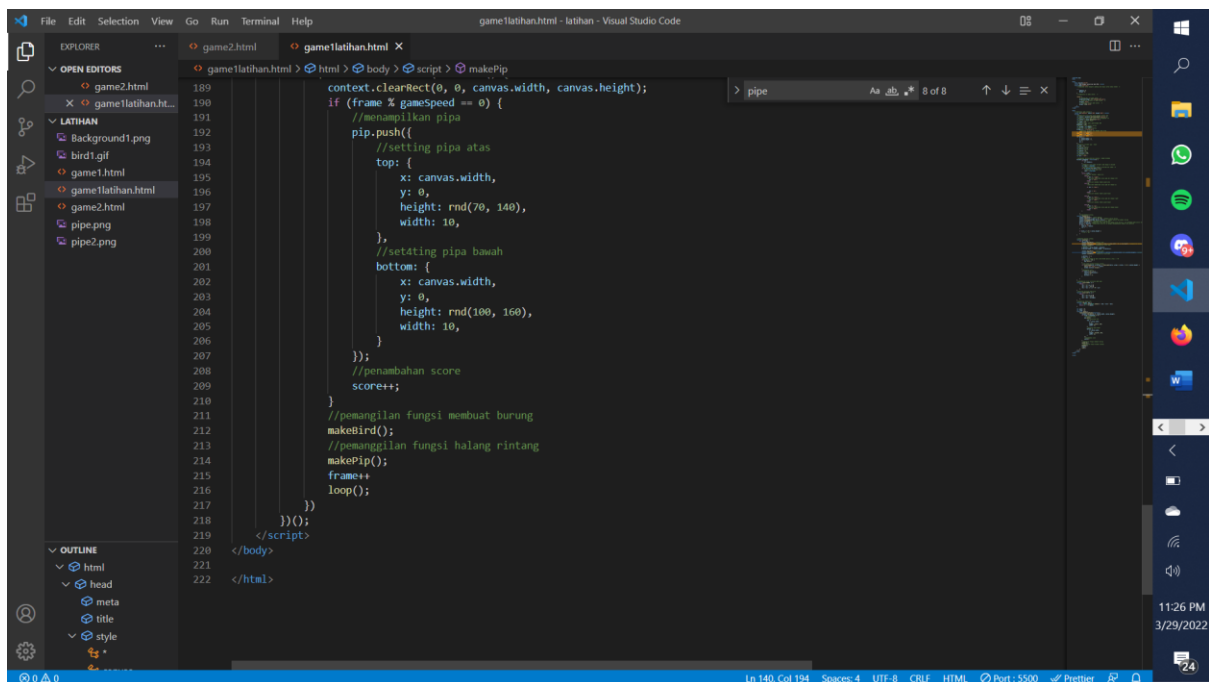
```
74 //pemberian nilai pada code
75 code = e.charCode;
76
77 switch (code) {
78   //jika user menekan panah kiri
79   case 37:
80     if (dir != 'right')
81       //maka akan memberikan nilai pada dir dengan left
82       dir = 'left';
83     break;
84   //jika user menekan tombol panah atas
85   case 38:
86     //maka akan memberikan nilai pada dir dengan up
87     if (dir != 'down')
88       dir = 'up';
89     break;
90   //jika user menekan tombol panah kanan
91   case 39:
92     if (dir != 'left')
93       //maka akan memberikan nilai pada dir dengan right
94       dir = 'right';
95     break;
96   //jika user menekan tombol panah bawah
97   case 40:
98     if (dir != 'up')
99       //maka akan memberikan nilai pada dir dengan bawah
100      dir = 'down';
101     break;
102 }
103
104 //fungsi membuat burung
105 function makeBird() {
106   context.beginPath(); //untuk memulai
107   context.rect(bird.x, bird.y, 10); //membuat ukuran burung
108 }
```



```
114 context.drawImage(birdg, bird.x, bird.y); // membuat posisi dan gambar
115 context.closePath(); //mengembalikan ke nilai awal
116 bird.y += bird.vy; //memberikan nilai pada bird.y dengan nilai bird.vy ini berdampak pada posisi ketinggian burung
117 bird.vy += gravity; //memberikan nilai bir.vy dengan menambahkannya dengan nilai gravity
118 if (dir == 'up') {
119   bird.vy = -bounce;
120   dir = '';
121 }
122
123 if (bird.y + 10 >= canvas.height) {
124   // dir = 'up'
125 }
126
127 //pembuatan halang rintang
128 function makePip() {
129   for (p of pip) {
130     context.beginPath(); //memulai path
131     context.drawImage(pipe1, p.top.x, p.top.y, p.top.width = 25, p.top.height); //pengaturan halang rintang
132     context.closePath(); //pengembalian titik awal
133     //setting jarak
134     p.bottom.y = p.top.height + distance;
135     p.bottom.height = (canvas.height - p.bottom.y);
136
137     context.beginPath(); //memulai path
138     context.drawImage(pipe2, p.bottom.x, p.bottom.y, p.bottom.width = 25, p.bottom.height); //pengaturan halang rintang
139     context.closePath(); // memulai di titik awal
140
141     p.bottom.x -= 1;
142     p.top.x -= 1;
143     //penghapuan obyek di awal jika berada posisi p.top.x < -10
144     if (p.top.x < -10) {
145       pip.shift();
146     }
147     //jika obyek terkena halang rintang
148     if (isCollide(bird, p.bottom) || isCollide(bird, p.top) || bird.y + 10 >= canvas.height) {
149       alert('Your score : ' + score);
150       window.location.reload();
151     }
152   }
153 }
```



```
153
154
155 //penambahan score
156 if (score % 10 == 0) {
157   console.log(distance);
158   score += 1;
159   distance -= 5;
160 }
161
162 //pengembalian nilai collision pada pipa
163 function isCollide0(b, p) {
164   return (
165     (b.x + 10 > p.x) &&
166     (b.x < p.x + 10) &&
167     (b.y > p.y || b.y + 10 > p.y)
168   );
169 }
170
171 //setting collision pada bird
172 function isCollide1(b, p) {
173   return (
174     (b.x + 10 > p.x) &&
175     (b.x < p.x + 10) &&
176     (b.y < p.y + p.height)
177   );
178 }
179
180 //pengaturan fungsi mtk
181 function rnd(min, max) {
182   var r = Math.round(Math.random() * (max - min) + min);
183   return (r - r % speed);
184 }
185
186 var score = 0;
187 var frame = 0;
188 (function loop() {
189   requestAnimationFrame(function() {
190     context.clearRect(0, 0, canvas.width, canvas.height);
191     if (frame % gameSpeed == 0) {
192       //menampilkan pipa
193       pip.push({
194         //setting pipa atas
195         top: {
196           x: canvas.width,
197           y: 0,
198           height: rnd(70, 140),
199           width: 10,
200         },
201         //setting pipa bawah
202         bottom: {
203           x: canvas.width,
204           y: 0,
205           height: rnd(100, 160),
206           width: 10,
207         }
208       });
209       //penambahan score
210       score++;
211     }
212     //pengisian fungsi membuat burung
213     makeBird();
214     //pengisian fungsi halang rintang
215     makePip();
216     frame++;
217     loop();
218   });
219 }());
220 </body>
221 </html>
```



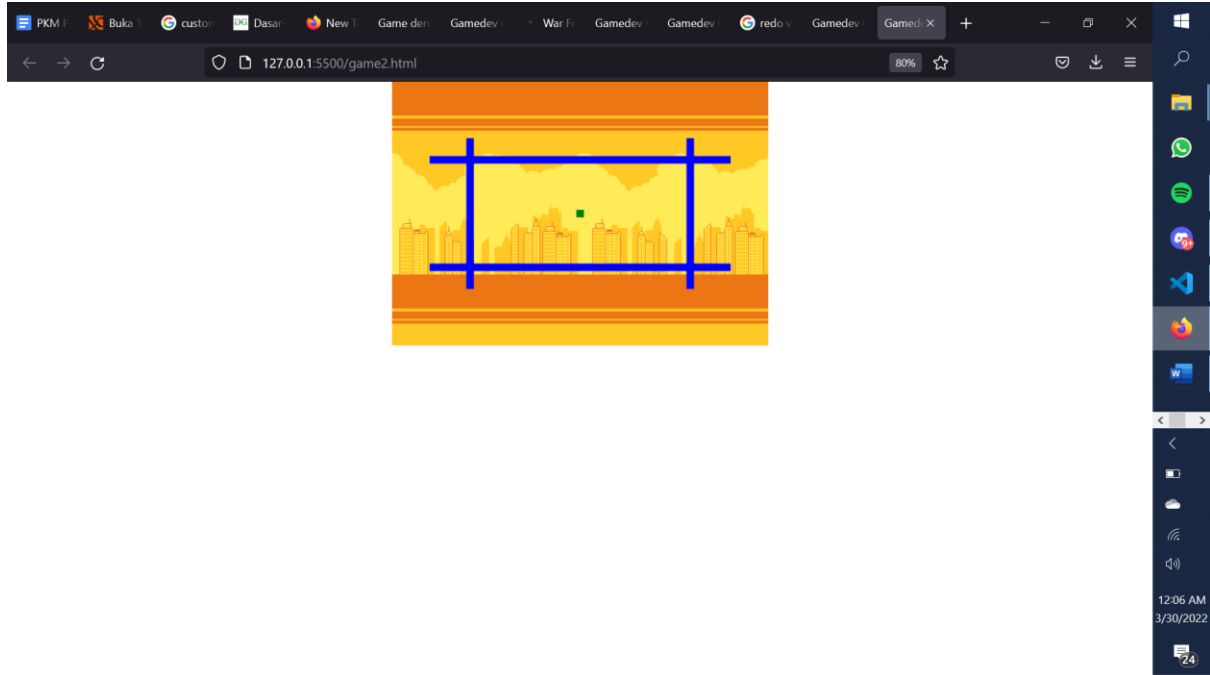
```
189
190 context.clearRect(0, 0, canvas.width, canvas.height);
191 if (frame % gameSpeed == 0) {
192   //menampilkan pipa
193   pip.push({
194     //setting pipa atas
195     top: {
196       x: canvas.width,
197       y: 0,
198       height: rnd(70, 140),
199       width: 10,
200     },
201     //setting pipa bawah
202     bottom: {
203       x: canvas.width,
204       y: 0,
205       height: rnd(100, 160),
206       width: 10,
207     }
208   });
209   //penambahan score
210   score++;
211 }
212 //pengisian fungsi membuat burung
213 makeBird();
214 //pengisian fungsi halang rintang
215 makePip();
216 frame++;
217 loop();
218 }());
219 </script>
220 </body>
221 </html>
```

Analisis :

Canvas pada html digunakan untuk sebagai wadah untuk menggambar grafik yang dilukis menggunakan code code javasript. Pada coding diatas element kanvas digunakan untuk membuat game fallpybird dengan menggunakan javascript. Untuk penjelasan koding diatas bisa dilihat melalui komentar komentar;

Game 2 :

Tampilan



Code:

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8" />
  <title>Gamedev Canvas Workshop</title>
  <!--Style pada css-->
  <style>
    /* pengaturan padding dan margin untuk semua komponen html */

    * {
      padding: 0;
      margin: 0;
    }
    /* pengaturan padding dan margin dan bakground untuk semua komponen
html */

    canvas {
      background: url("Background1.png");
      display: block;
      margin: 0 auto;
    }
  </style>
</head>

<body>
  <!--Pembuatan elemn kanvas-->
```

```

<canvas id="myCanvas" width="500" height="350"></canvas>
<script>
    //pemberian nilai canvas dengan penyelesaian dom dari id my canvas
    var canvas = document.getElementById('myCanvas');
    //membbberikan akses pada canvas untuk menggambar 2d
    var context = canvas.getContext('2d');
    var speed = 10; //dekrarasi variabel speed
    var gameSpeed = 1; //deklarasi game speed
    //pembuatan variabel me dengan type array sebagai objek utama
    var me = {
        x: canvas.width / 2 - 5,
        y: canvas.height / 2 - 5,
        width: 10,
        height: 10
    }

    var bullets = []; //deklarasi variable bullets
    var wallLength = 200; //deklarasi variable length untuk anjang dinding
rintang
    //pembuatan obyek wall dengan variablewalls bertipe array
    var walls = {
        //pembuatan walls sebelah kiri
        'left': {
            x: 10,
            y: canvas.height / 2 - wallLength / 2,
            width: 10,
            height: wallLength,
            'pos': 'left'
        },
        //pembuatan walls sebelah kanan
        'right': {
            x: canvas.width - 20,
            y: canvas.height / 2 - wallLength / 2,
            width: 10,
            height: wallLength,
            'pos': 'right'
        },
        //pembuatan walls sebelah atas
        'top': {
            x: canvas.width / 2 - wallLength,
            y: 10,
            width: wallLength * 2,
            height: 10,
            'pos': 'top'
        },
        //pembuatan walls sebelah bawah
        'bottom': {

```

```

        x: canvas.width / 2 - wallLength,
        y: canvas.height - 20,
        width: wallLength * 2,
        height: 10,
        'pos': 'bottom'
    },
};
//deklarasi dan inisialisasi gravitasi me
var gravity = 0.2;
//deklarasi dan inisialisasi pantulan
var bounce = 4;
//deklarasi dan inisialisasi jarak
var distance = 70;
var dir = '';
//pengaturan aksi ketika user menekan sebuah tombol
document.onkeyup = function(e) {
    if (!e) {
        e = window.e;
    }
    var code = e.keyCode;
    if (e.charCode && codee == 0) {
        code = e.charCode;
    }
    //pengecekan inputan user
    switch (code) {
        //ketika menekan arrow kiri
        case 37:
            if (dir != 'left')
                dir = 'left';
            //pengaturan dinding ketika terkena bullet maka akan
            bergeser 10px ke kakiri
            bullets.push({
                x: me.x,
                y: me.y,
                width: 10,
                height: 10,
                dir: dir
            });
            break;
            //ketika menekan arrow bawah
        case 40:
            if (dir != 'down')
                dir = 'down';
            //pengaturan dinding ketika terkena bullet maka akan
            bergeser 10px ke kabawah
            bullets.push({
                x: me.x,
                y: me.y,

```

```

        width: 10,
        height: 10,
        dir: dir
    });
    break;
    //ketika menekan arrow kanan
case 39:
    if (dir !== 'right')
        dir = 'right';
    //pengaturan dinding ketika terkena bullet maka akan
    bergeser 10px ke kanan
    bullets.push({
        x: me.x,
        y: me.y,
        width: 10,
        height: 10,
        dir: dir
    });
    break;
    //ketika menekan arrow atas
case 38:
    if (dir !== 'up')
        dir = 'up';
    //pengaturan dinding ketika terkena bullet maka akan
    bergeser 10px ke kaatas
    bullets.push({
        x: me.x,
        y: me.y,
        width: 10,
        height: 10,
        dir: dir
    });
    break;
    }
}
//deklarasi dan inisialisasi wallspeed
var wallSpeed = 0.2;
//deklarasi dan inisialisasi wallet backward (digunakan untuk bullet
ketika terkena dinding)
var walletBackward = 20;
//deklarasi dan inisialisasi score
var score = 0;
//pembuatan walls
function makeWalls() {
    //looping untuk dinding tetap bergerak
    for (var index in walls) {
        var wall = walls[index];

```



```

        context.beginPath(); //pemanggilan fungsi begin path
untukmemulai
        context.rect(wall.x, wall.y, wall.width, wall.height);
//inisialisasi ukuran dinding
        context.fillStyle = "blue"; //inisialisasi warna dinding
        context.fill(); //pemberian warna pada dinding
        context.closePath(); //pengembalian pada titik awal
        if (wall.pos == 'left') {
            wall.x += wallSpeed;
            //pengecekan apakah me tersentuh dinding jika iya maka
akan menampilkan score
            if ((wall.x + wall.width) >= me.x) {
                loose();
            }
        }
        if (wall.pos == 'right') {
            wall.x -= wallSpeed;
            //pengecekan apakah me tersentuh dinding jika iya maka
akan menampilkan score
            if ((wall.x + wall.width) <= me.x) {
                loose();
            }
        }
        if (wall.pos == 'top') {
            wall.y += wallSpeed;
            //pengecekan apakah me tersentuh dinding jika iya maka
akan menampilkan score
            if ((wall.y + wall.height) >= me.y) {
                loose();
            }
        }
        if (wall.pos == 'bottom') {
            wall.y -= wallSpeed;
            //pengecekan apakah me tersentuh dinding jika iya maka
akan menampilkan score
            if ((wall.y + wall.height) <= me.y) {
                loose();
            }
        }

        console.log(walls[wall]) //menampilkan wall pada kanvas

    }
}
//fungsi ketika game berakhir maka akan menampilkan alert score
function loose() {
    alert('You Lose! score : ' + score);
}

```

```

var collid = false;

function makeBullet() {
    for (bullet of bullets) {
        context.beginPath(); //mulai
        context.rect(bullet.x, bullet.y, bullet.width, bullet.height);
//seting ukuran peluru
        context.fillStyle = "red"; //pemberian warna red
        context.fill(); //pengisian warna
        context.closePath(); //pengembalian warna
        if (bullet.dir == 'left') {
            bullet.x -= 10;
        }
        if (bullet.dir == 'right') {
            bullet.x += 10;
        }
        if (bullet.dir == 'up') {
            bullet.y -= 10;
        }
        if (bullet.dir == 'down') {
            bullet.y += 10;
        }
        if (bullet.x >= canvas.width || bullet.x <= 0 || bullet.y >=
canvas.height || bullet.y <= 0) {
            bullets.splice(bullets.indexOf(bullet), 1);
        }
        //pemberian perintah ketika perulu menyentuh dinding kanan
maka dinding bergeser ke kananC
        if (bullet.dir == 'right') {
            if (bullet.x >= walls['right'].x) {
                walls['right'].x = walls['right'].x + walletBackward;
                collid = true;
            }
        }
        //pemberian perintah ketika perulu menyentuh dinding atas maka
dinding bergeser ke atas
        if (bullet.dir == 'up') {
            if (bullet.y <= walls['top'].y) {
                walls['top'].y = walls['top'].y - walletBackward;
                collid = true;
            }
        }
        //pemberian perintah ketika perulu menyentuh dinding bawah
maka dinding bergeser ke bawah
        if (bullet.dir == 'down') {
            if (bullet.y >= walls['bottom'].y) {
                walls['bottom'].y = walls['bottom'].y +
walletBackward;

```

```

        collid = true;
    }
}
//pemberian perintah ketika perulu menyentuh dinding kiri maka
dinding bergeser ke kiri
if (bullet.dir == 'left') {
    if (bullet.x <= walls['left'].x) {
        walls['left'].x = walls['left'].x - walletBackward;
        collid = true;
    }
}
//untuk menambahkan skor
if (collid) {
    bullets.splice(bullets.indexOf(bullet), 1);
    collid = false;
    score++;
}

}
}
//function menghapus bullet
function removeBullet(index) {
    bullets.splice(index, 1);
}
//setiing onjek kotak utama
function makeMe() {
    context.beginPath(); //mulai metode
    context.rect(me.x, me.y, me.width, me.height); //mengatur ukuran
    context.fillStyle = "green"; // inisialisasi warna hijau
    context.fill(); //pemberian warna
    context.closePath(); //pengembailian ke titik awal
}
//perhitungan r
function rnd(min, max) {
    var r = Math.round(Math.random() * (max - min) + min);
    return (r - r % speed);
}
var wallPoses = ['left', 'right', 'top', 'bottom'];
var frame = 0;
//mengeksekusi dan menambah kecepatan jika score semakin bertambah
(function loop() {
    requestAnimationFrame(function() {
        context.clearRect(0, 0, canvas.width, canvas.height);
        makeMe(); //pemanggilan objek utama
        makeBullet(); //pemanggilan peluru
        makeWalls(); //pemannggilan dinding
        if (score % 50 == 0) {

```

```

        walletBackward -= 1; //pengurangan jarak dinding setelah
ditembak

        wallSpeed += 0.1; //penambahan speed dinding
        score += 1; //penambahan score
        console.log(score); //menampilkan score pemain
        console.log(wallSpeed) //menampilkan dinding
        console.log(walletBackward); //menampilkan dinding ketika
terkena peluru
    }
    //jika score bertambah kecepatan dinding bertambah
    if (frame % gameSpeed == 0) {
        frame++
    }
    loop();
}
})();
</script>
</body>

</html>

```

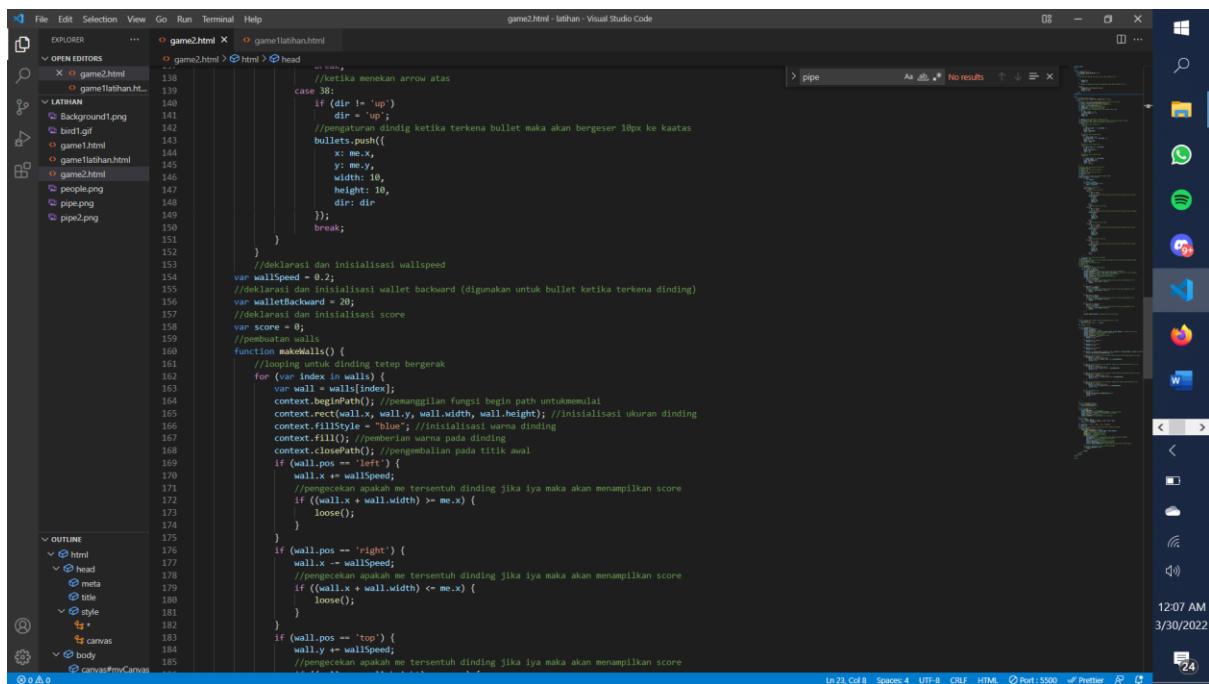
```

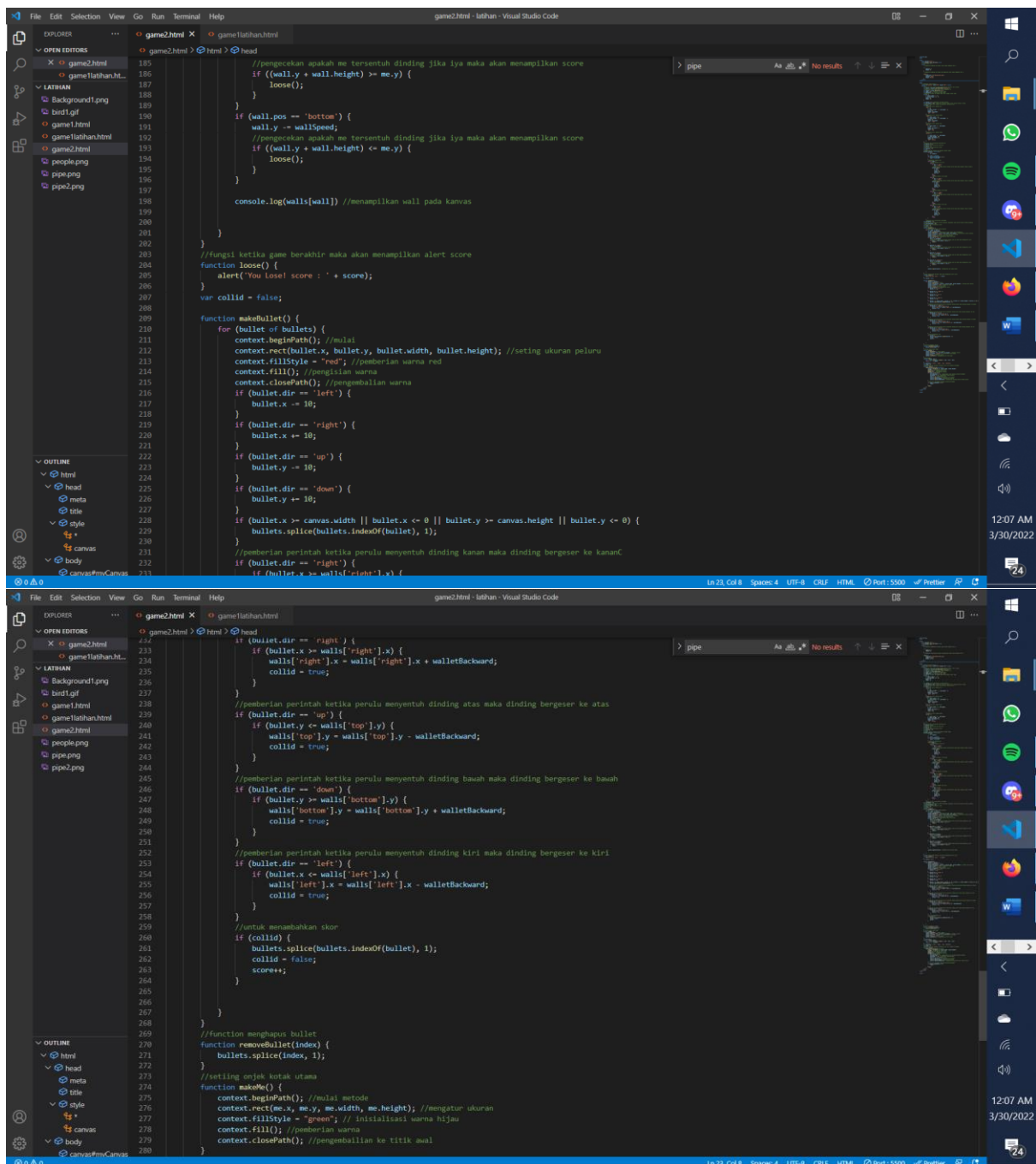
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5      <meta charset="utf-8" />
6      <title>Gamedev Canvas Workshop</title>
7      <!--Style pada css-->
8      <style>
9          /* pengaturan padding dan margin untuk semua komponen html */
10
11          * {
12              padding: 0;
13              margin: 0;
14          }
15          /* pengaturan padding dan margin dan background untuk semua komponen html */
16
17          canvas {
18              background: url("Background1.png");
19              display: block;
20              margin: 0 auto;
21          }
22      </style>
23  </head>
24
25  <body>
26      <!--Pembuatan elemen canvas-->
27      <canvas id="myCanvas" width="500" height="350"></canvas>
28      <script>
29          //pemberian nilai canvas dengan penyelesaian dom dari id my canvas
30          var canvas = document.getElementById('myCanvas');
31          //memberikan akses pada canvas untuk menggambar 2d
32          var context = canvas.getContext('2d');
33          var speed = 10; //deklarasi variabel speed
34          var gameSpeed = 1; //deklarasi game speed
35          //pembuatan variabel me dengan type array sebagai objek utama
36          var me = {
37              x: canvas.width / 2 - 5,
38              y: canvas.height / 2 - 5,
39              width: 10,
40              height: 10
41          }
42
43          var bullets = []; //deklarasi variabels bullets
44          var walllength = 200; //deklarasi variabel length untuk anjang dinding rintang
45          //pembuatan objek wall dengan variabelwalls bertipe array
46          var walls = [
47              //pembuatan walls sebelah kiri

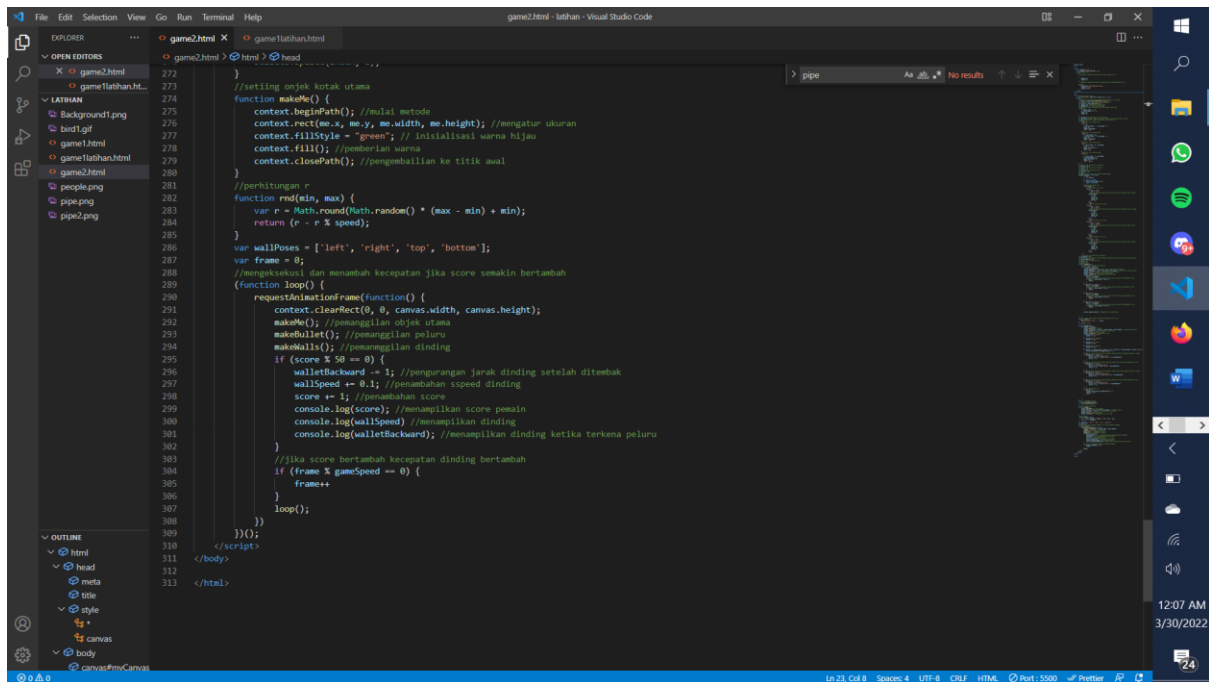
```

```
47 //pembuatan walli sebelah kiri
48 'left': {
49   x: 10,
50   y: canvas.height / 2 - walllength / 2,
51   width: 10,
52   height: walllength,
53   'pos': 'left'
54 },
55 //pembuatan walli sebelah kanan
56 'right': {
57   x: canvas.width - 20,
58   y: canvas.height / 2 - walllength / 2,
59   width: 10,
60   height: walllength,
61   'pos': 'right'
62 },
63 //pembuatan walli sebelah atas
64 'top': {
65   x: canvas.width / 2 - walllength,
66   y: 10,
67   width: walllength * 2,
68   height: 10,
69   'pos': 'top'
70 },
71 //pembuatan walli sebelah bawah
72 'bottom': {
73   x: canvas.width / 2 - walllength,
74   y: canvas.height - 20,
75   width: walllength * 2,
76   height: 10,
77   'pos': 'bottom'
78 },
79
80 };
81 //deklarasi dan inialisasi gravitasi me
82 var gravity = 0.2;
83 //deklarasi dan inialisasi pantulan
84 var bounce = 4;
85 //deklarasi dan inialisasi jarak
86 var distance = 70;
87 var dir = '';
88 //pengaturan aksi ketika user menekan sebuah tombol
89 document.onkeyup = function(e) {
90   if (ie) {
91     e = window.e;
92   }
93   var code = e.keyCode;
94   if (e.charCode && code == 0) {
95     code = e.charCode;
96   }
97   //pengecekan inputan user
98   switch (code) {
99     //ketika menekan arrow kiri
100     case 37:
101       if (dir != 'left')
102         dir = 'left';
103       //pengaturan dinding ketika terkena bullet maka akan bergeser 10px ke kakiri
104       bullets.push({
105         x: me.x,
106         y: me.y,
107         width: 10,
108         height: 10,
109         dir: dir
110       });
111       break;
112     //ketika menekan arrow bawah
113     case 40:
114       if (dir != 'down')
115         dir = 'down';
116       //pengaturan dinding ketika terkena bullet maka akan bergeser 10px ke kawah
117       bullets.push({
118         x: me.x,
119         y: me.y,
120         width: 10,
121         height: 10,
122         dir: dir
123       });
124       break;
125     //ketika menekan arrow kanan
126     case 39:
127       if (dir != 'right')
128         dir = 'right';
129       //pengaturan dinding ketika terkena bullet maka akan bergeser 10px ke kanan
130       bullets.push({
131         x: me.x,
132         y: me.y,
133         width: 10,
134         height: 10,
135         dir: dir
136       });
137       break;
138     //ketika menekan arrow atas
139     case 38:
140       if (dir != 'up')
141         dir = 'up';
142       //pengaturan dinding ketika terkena bullet maka akan bergeser 10px ke kaatas
```

```
94 if (e.charCode && code == 0) {
95   code = e.charCode;
96 }
97 //pengecekan inputan user
98 switch (code) {
99   //ketika menekan arrow kiri
100   case 37:
101     if (dir != 'left')
102       dir = 'left';
103     //pengaturan dinding ketika terkena bullet maka akan bergeser 10px ke kakiri
104     bullets.push({
105       x: me.x,
106       y: me.y,
107       width: 10,
108       height: 10,
109       dir: dir
110     });
111     break;
112   //ketika menekan arrow bawah
113   case 40:
114     if (dir != 'down')
115       dir = 'down';
116     //pengaturan dinding ketika terkena bullet maka akan bergeser 10px ke kawah
117     bullets.push({
118       x: me.x,
119       y: me.y,
120       width: 10,
121       height: 10,
122       dir: dir
123     });
124     break;
125   //ketika menekan arrow kanan
126   case 39:
127     if (dir != 'right')
128       dir = 'right';
129     //pengaturan dinding ketika terkena bullet maka akan bergeser 10px ke kanan
130     bullets.push({
131       x: me.x,
132       y: me.y,
133       width: 10,
134       height: 10,
135       dir: dir
136     });
137     break;
138   //ketika menekan arrow atas
139   case 38:
140     if (dir != 'up')
141       dir = 'up';
142     //pengaturan dinding ketika terkena bullet maka akan bergeser 10px ke kaatas
```







Analisis :

Canvas pada html digunakan untuk sebagai wadah untuk menggambar grafik yang dilukis menggunakan code code javasript. Pada coding diatas element kanvas digunakan untuk membuat game fallpybird dengan menggunakan javascript. Untuk penjelasan koding diatas bisa dilihat melalui komentar komentar.;

KESIMPULAN

Elemen "kanvas" HTML digunakan untuk menggambar grafik melalui JavaScript. Elemen "kanvas" hanyalah wadah untuk grafik. Kanvas memiliki beberapa metode untuk menggambar jalur, kotak, lingkaran, teks, dan menambahkan gambar. Kanvas akan menjadi area persegi panjang pada halaman HTML. Secara default, kanvas tidak memiliki batas dan tidak ada konten. Canvas pada HTML dilukis dengan menggunakan code javascript dimana Properti yang Didukung: Properti seperti Colors, Styles, Shadows, Line Styles, Rectangles, Paths, Transformations, Text, Pixel Manipulation, Compositing & Image Drawing, adalah atribut global yang didukung oleh semua tag kanvas