



ADVENTURE OF DINACOM

Phaser

Adalah sebuah framework yang dapat membantu kita untuk mengembangkan sebuah game berbasis web dengan menggunakan bahasa JavaScript dan TypeScript.

phaser.io || examples.phaser.io || docs.phaser.io

State Management

Adalah sebuah metode penyusunan kode agar lebih terstruktur dan terarah.

Tutorial Pembuatan Adventure Of Dinacom

Struktur Awal (index.html)

```
<!DOCTYPE html>
<html>
<head>
  <title>Adventure of Dinacom</title>
  <script type="text/javascript" src="assets/js/phaser.min.js"></script>
  <script type="text/javascript" src="assets/js/starting.js"></script>
  <style type="text/css">
    *{margin: 0; padding: 0;}
  </style>
  <script type="text/javascript">
    (function() {
      if(localStorage.highScore && localStorage.nama){
        localStorage.highScore = localStorage.highScore;
        localStorage.nama = localStorage.nama;
      } else {
        localStorage.highScore = "";
        localStorage.nama = "";
      }
      var game = new Phaser.Game(1365, 650, Phaser.AUTO, "advdinacom");
      game.state.add("startng", startng);
      game.state.start("startng");
    })();
  </script>
</head>
<body>
  <div id="advdinacom"></div>
</body>
</html>
```

DINACOM.DINUS.AC.ID



dinacom@dinus.id



[fb.com/dinacomOfficial](https://www.facebook.com/dinacomOfficial)



[@dinacomOfficial](https://twitter.com/dinacomOfficial)



Pada halaman index terdapat perintah `game.state.start("starting");` yang berarti bahwa perintah tersebut memanggil fungsi `starting`. Pada fungsi `preload` yang ada di dalam state dari `starting` memload asset gambar. Dan selanjutnya membuat tampilan untuk penayangan asset dari `"img_load"` dengan perintah.

`starting.js (assets/js/starting.js)`

```
var starting = function(game){
    //console.log digunakan untuk memberitanda apakah fungsi di file sudah benar
    console.log("%cStarting", "color:#FFFFFF; background:#FF0000;");
}

starting.prototype = {
    preload: function(){
        // memload gambar loading
        this.game.load.image("img_load", "assets/img/loading.gif");
    },
    create: function(){
        this.scale.scaleMode = Phaser.ScaleManager.SHOW_ALL;
        // membuat mode agar efeknya terlihat dari keseluruhan

        this.scale.pageAlignHorizontally = true;
        // mengkondisikan bahwa halaman akan terbuka secara horizontal

        this.scale.setScreenSize();
        // untuk mengeset ukuran screen mengikuti ukuran dari frame

        this.game.state.start("loading");
        // memanggil state loading
    }
}
```

Selanjutnya di file `starting.js` akan memanggil state **loading**. Pada state `loading` ini digunakan untuk memload semua asset yang akan diperlukan pada game nanti seperti gambar, map, sound.

Tambahkan pemanggilan file `loading.js` di `index.html` setelah pemanggilan file `phaser.min.js`. setelah itu buat file `loading.js` dan save di `assets/js/loading.js`

DINACOM.DINUS.AC.ID



dinacom@dinus.id



fb.com/dinacomOfficial

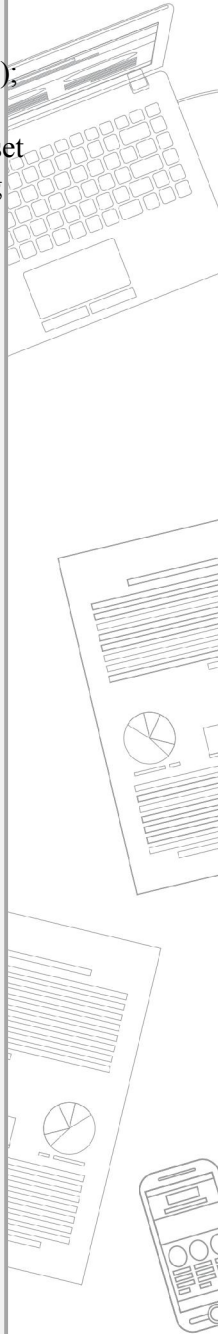


[@dinacomOfficial](https://twitter.com/dinacomOfficial)



```
loading.js (assets/js/loading.js)
var loading = function(game) {
    console.log("%cLoading", "color:white; background:#97c5c7;");
}

loading.prototype = {
    preload: function() {
        //menampilkangambarimg_load
        var loading = this.add.sprite(this.game.width/2, this.game.height/2, 'img_load');
        loading.anchor.setTo(0.5, 0.5);
        this.load.setPreloadSprite(loading); //file loading akanberjalansesuai load asset
        this.game.stage.backgroundColor = "#97c5c7"; //menset background loading
        this.game.load.tilemap('mapdinacom', 'assets/img/mapdinacom.json', null,
        Phaser.Tilemap.TILED_JSON);
        //load file gambarpecahan
        this.game.load.spritesheet('playeradv', 'assets/img/dude1.png', 43.5, 53);
        //load gambar yang diperlukan
        this.game.load.image('obj1', 'assets/img/obj1.png');
        this.game.load.image('tanah', 'assets/img/tanah.png');
        this.game.load.image('virus', 'assets/img/virus.png');
        this.game.load.image('bgcover', 'assets/img/background_cover.png');
        this.game.load.image('cover', 'assets/img/cover.png');
        this.game.load.image('smallcoin', 'assets/img/koin.png');
        this.game.load.image('boss', 'assets/img/boss.png');
        this.game.load.image('shot', 'assets/img/shot.png');
        this.game.load.image('full', 'assets/img/full.png');
        this.game.load.image('bigcoin', 'assets/img/koin1.png');
        this.game.load.image('prince', 'assets/img/prince.png');
        this.game.load.image('background', 'assets/img/background.png');
        this.game.load.image('img_title', 'assets/img/title.png');
        this.game.load.image('img_over', 'assets/img/gameover.png');
        this.game.load.image('img_tombolstart', 'assets/img/tombolstart.png');
        this.game.load.image('img_tombolmainlagi', 'assets/img/playagain.png');
        this.game.load.image('img_icon', 'assets/img/icon.png');
        //meload sound
        this.game.load.audio('audio_duar', ['assets/sounds/aduar.wav']);
        this.game.load.audio('audio_cling', ['assets/sounds/cling.wav']);
        this.game.load.audio('audio_over', ['assets/sounds/over.wav']);
    },
    create: function() {
        this.game.state.start("cover");
    }
}
```



DINACOM.DINUS.AC.ID



dinacom@dinus.id



fb.com/dinacomOfficial



[@dinacomOfficial](https://twitter.com/dinacomOfficial)



Fungsi create pada loading.js digunakan untuk memulai state **cover**, state cover ini berisi cover game atau menu game sebelum memulai game.

cover.js (assets/js/cover.js)

```
var cover = function(game){
    console.log("%cCover", "color:white; background:green");
}

cover.prototype = {
    create: function(){
        // Menampilkan gambar background
        this.background = this.game.add.tileSprite(0, 0, 1365, 650, "bgcover");
        // auto scroll digunakan untuk menggerakkan gambar secara otomatis
        this.background.autoScroll(-90, 0);
        // untuk menampilkan gambar cover, icon, title
        this.game.add.sprite(0, 0, 'cover');
        this.judul = this.game.add.sprite(760, 400, 'img_title');
        this.icon = this.game.add.sprite(590, 80, 'img_icon');
        // membuat animasi tween (naik turun)
        this.game.add.tween(this.icon).to({y: 100}, 1000, Phaser.Easing.Linear, true, 0, 100, true);
        // menambahkan tombol start dengan action untuk menjalankan state main
        var tombol = this.game.add.button(this.game.width/2, 400, "img_tombolstart", this.mainkan, this);
        tombol.anchor.setTo(0.5, 0.5);

        // PLAY SOUND EFFECT
        this.sound_awal = this.game.add.audio('audio_duar');
        this.sound_awal.play();
    },
    mainkan: function(){
        this.game.state.start("main");
    }
}
```

Pada fungsi mainkan didalam fungsi tersebut memanggil dan menjalankan state **main**, dimana pada state **main** ini adalah inti utama dari game yang akan kita buat. Didalam main terdapat aturan saat tombol keyboard ditekan saat karakter utama atau player menabrak musuh atau jatuh dan lain sebagainya akan kita bangun didalam state **main**. Langsung saja berikut source code nya :

DINACOM.DINUS.AC.ID



dinacom@dinus.id



fb.com/dinacomOfficial



[@dinacomOfficial](https://twitter.com/dinacomOfficial)



Main.js(assets/js/main.js)

```
var main = function(game){
  console.log("%cMainGame", "color:white; background:orange");
}
main.prototype = {

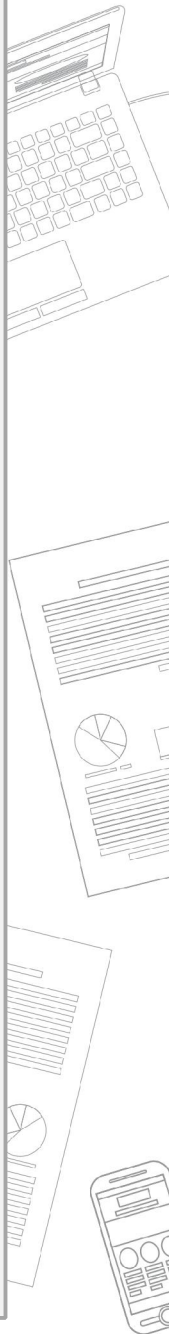
  create:function (){
    this.score = 0;
    this.facing = 'left';
    this.jumpTimer = 0;
    this.bosshot = 0;
    this.bosjump = 0;
    /* ----- PENDEVINISIAN MEDAN GAME ----- */
    this.game.physics.startSystem(Phaser.Physics.ARCADE);
    this.game.stage.backgroundColor = '#000000'; //background color
    this.bg = this.game.add.tileSprite(0, 0, 1366, 650, 'background'); //background gambar
    this.bg.fixedToCamera = true; //agar background mengikuti kamera

    this.map = this.game.add.tilemap('mapdinacom'); //menambahkan tile map
    this.map.addTilesetImage('obj1'); //source gambar tilemap
    this.map.addTilesetImage('tanah'); //source gambar tilemap
    this.map.setCollisionByExclusion([]); //pengecualian collide di maps dari index tile
    this.layer = this.map.createLayer('Adventure Dinacom'); //layer map
    this.layer.resizeWorld(); //layer menyesuaikan canvas

    /* ----- PENDEVINISIAN PLAYER, MUSUH DAN BOS ----- */
    this.player = this.game.add.sprite(32, 1810, 'playeradv'); //menambahkan player
    this.prince = this.game.add.sprite(2075, 1836, 'prince'); //menambahkan prince
    this.enemies = this.game.add.sprite(809, 1685, 'virus'); //menambahkan musuh 1
    this.enemiess = this.game.add.sprite(1453, 1770, 'virus'); //menambahkan musuh 2
    this.bos = this.game.add.sprite(2130, 1741, 'boss'); //menambahkan bos
    /* END PENDEVINISIAN */

    /* PENGAKTIFAN PHASER PHYSICS TIAP KARAKTER */
    this.game.physics.enable(this.bos, Phaser.Physics.ARCADE);
    this.game.physics.enable(this.prince, Phaser.Physics.ARCADE);
    this.game.physics.enable(this.player, Phaser.Physics.ARCADE);
    this.game.physics.enable(this.enemies, Phaser.Physics.ARCADE);
    this.game.physics.enable(this.enemiess, Phaser.Physics.ARCADE);
    /* END PHYSICS */

    /* atribut bos */
    this.bos.body.gravity.y=300;
    this.shots = this.game.add.group(); //PENDEVINISIAN UNTUK GROUP TEMBAKAN
    this.shots.enableBody = true; //MENGAKTIFKAN TUBUH
    this.shots.physicsBodyType = Phaser.Physics.ARCADE; //MENGAKTIFKAN PHYSICS
  }
```



DINACOM.DINUS.AC.ID



dinacom@dinus.id



fb.com/dinacomOfficial



[@dinacomOfficial](https://twitter.com/dinacomOfficial)



Lanjutan dari main.js

```
/* ATRIBUT KARAKTER MUSUH 2 */
this.enemies.body.collideWorldBounds = true; //tidak bisa keluar canvas
this.enemies.body.gravity.y=250; //mengatur gravitasi musuh
this.enemies.body.velocity.x=100; //kecepatan musuh
this.enemies.animations.play('left',20,true); //play animasi kiri
/* END KARAKTER*/

/* ATRIBUT KARAKTER MUSUH 2 */
this.enemies.body.collideWorldBounds = true;
this.enemies.body.gravity.y=500;
this.enemies.body.velocity.x=100;
this.enemies.animations.play('left',20,true);
/* END KARAKTER*/

/* ATRIBUT KARAKTER PLAYER */
this.player.body.bounce.y = 0.2; //MANTUL SAAT SENTUH BAWAH
this.player.checkWorldBounds = true; //MENGECEK PLAYER KELUAR CANVAS ATAU TIDAK
this.player.events.onOutOfBounds.add(this.gameover,this); //JIKA KELUAR MAKA AKAN GAME
OVER
this.player.body.setSize(27, 35, 10, 16);
this.player.body.gravity.y=350;
this.player.animations.add('left', [0, 1, 2, 3], 10, true);
this.player.animations.add('turn', [4], 20, true);
this.player.animations.add('right', [5, 6, 7, 6], 10, true);

this.game.camera.follow(this.player); //CAMERA GAME MENGIKUTI PLAYER
/* END KARAKTER*/

/* PENAMBAHAN KOIN */
this.star1 = this.game.add.sprite(495.50,1773,'smallcoin'); //MENAMBAHKAN SPRITE
this.star1.enableBody = true; //MENGAKTIFKAN TUBUH
this.game.physics.arcade.enable(this.star1,Phaser.Physics.ARCADE); //AKTIFKAN PHYSICS
this.star1.body.gravity.y=0; //GRAVITY

this.star2 = this.game.add.sprite(809,1685,'smallcoin');
this.star2.enableBody = true;
this.game.physics.arcade.enable(this.star2,Phaser.Physics.ARCADE);
this.star2.body.gravity.y=0;

this.star3 = this.game.add.sprite(1150,1780,'smallcoin');
this.star3.enableBody = true;
this.game.physics.arcade.enable(this.star3,Phaser.Physics.ARCADE);
this.star3.body.gravity.y=0;

this.star4 = this.game.add.sprite(1453,1770,'smallcoin');
this.star4.enableBody = true;
this.game.physics.arcade.enable(this.star4,Phaser.Physics.ARCADE);
this.star4.body.gravity.y=0;

this.bigstar1 = this.game.add.sprite(1695,1549,'bigcoin');
this.bigstar1.enableBody = true;
this.game.physics.arcade.enable(this.bigstar1,Phaser.Physics.ARCADE);
this.bigstar1.body.gravity.y=0;
/* END KOIN */
```

DINACOM.DINUS.AC.ID



dinacom@dinus.id



[fb.com/dinacomOfficial](https://www.facebook.com/dinacomOfficial)



[@dinacomOfficial](https://twitter.com/dinacomOfficial)



Lanjutan dari main.js(function create dan update)

```
/* KEPERLUAN CONTROL */
this.cursors = this.game.input.keyboard.createCursorKeys(); //KEYBOARD ARAH
this.jumpButton = this.game.input.keyboard.addKey(Phaser.Keyboard.SPACEBAR); //SPACE
this.aduar = this.game.add.audio('audio_cling'); //PENDEKINISAN AUDIO

/* PENAMBAHAN TULISAN SCORE */
this.tulisanscore = this.game.add.text(16, 16, 'Score : ' + this.score, { fill: '#ffffff' }); //MENAMBAHKAN
TEXT
this.tulisanscore.fixedToCamera = true; //TULISAN FIX KE CAMERA TIDAK PINDAH PINDAH
POSISI
this.tulisanscore.cameraOffset.x = 10; //PENYESUAIAN DI KAMERA X
this.tulisanscore.cameraOffset.y = 20; //PENYESUAIAN DI KAMERA Y
}, /*Akhir dari function create pada main.js */

update:function() {
    this.tulisanscore.text = 'Score : ' + this.score; //UPDATE TULISAN SCORE
    /* EVENT SENTUH MENYENTUH ANTAR KARAKTER DAN LAYER */
    this.game.physics.arcade.collide(this.player, this.layer); //PLAYER NPAK DI LAYER
    this.game.physics.arcade.collide(this.bos, this.layer); //BOS NPAK DI LAYER
    //JIKA PLAYER BERSENTUHAN TEMBAKAN AKAN MEMANGGIL FUNGSI GAME OVER
    this.game.physics.arcade.collide(this.player, this.shots,this.gameover,null,this);
    //JIKA PLAYER BERSENTUHAN BOS AKAN MEMANGGIL FUNGSI GAME OVER
    this.game.physics.arcade.collide(this.player, this.bos,this.gameover,null,this);
    //JIKA MUSUH 1 BERSENTUHAN DENGAN LAYER AKAN MEMANGGIL FUNGSI CEK
    this.game.physics.arcade.collide(this.enemies, this.layer,this.cek,null,this);
    //JIKA MUSUH 2 BERSENTUHAN DENGAN LAYER AKAN MEMANGGIL FUNGSI CEK
    this.game.physics.arcade.collide(this.enemies, this.layer,this.cek,null,this);
    //JIKA PLAYER BERSENTUHAN DENGAN MUSUH 1 AKAN MEMANGGIL FUNGSI GAME OVER
    this.game.physics.arcade.collide(this.player, this.enemies,this.gameover,null,this);
    //JIKA PLAYER BERSENTUHAN DENGAN MUSUH 2 AKAN MEMANGGIL FUNGSI GAME OVER
    this.game.physics.arcade.collide(this.player, this.enemies, this.gameover,null,this);
    //JIKA PLAYER BERSENTUHAN DENGAN PRINCE AKAN MEMANGGIL FINISH
    this.game.physics.arcade.overlap(this.player, this.prince, this.finish,null,this);
    //JIKA PLAYER BERSENTUHAN DENGAN KOIN 1 AKAN MEMANGGIL TAMBAH
    this.game.physics.arcade.overlap(this.player, this.star1, this.tambah, null, this);
    //JIKA PLAYER BERSENTUHAN DENGAN KOIN 2 AKAN MEMANGGIL TAMBAH
    this.game.physics.arcade.overlap(this.player, this.star2, this.tambah, null, this);
    //JIKA PLAYER BERSENTUHAN DENGAN KOIN 3 AKAN MEMANGGIL TAMBAH
    this.game.physics.arcade.overlap(this.player, this.star3, this.tambah, null, this);
    //JIKA PLAYER BERSENTUHAN DENGAN KOIN 3 AKAN MEMANGGIL TAMBAH
    this.game.physics.arcade.overlap(this.player, this.star4, this.tambah, null, this);
    //JIKA PLAYER BERSENTUHAN DENGAN KOIN BESAR AKAN MEMANGGIL TAMBAH
    this.game.physics.arcade.overlap(this.player, this.bigstar1, this.tambah, null, this);
    //JIKA WAKTU GAME SEKARANG LEBIH BESAR DARI BOSS SHOT
    if(this.game.time.now > this.bosshot){
        this.shot = this.shots.create(this.bos.x,this.bos.y,'shot'); //MEMANGGIL PELURU
        this.shot.body.velocity.x = -100; //KECEPATAN PLURU
        this.bosshot = this.game.time.now + 2000; //BOSSSHOT WAKTU GAME SAAT INI + 2000
        BERTUJUAN MEMBERI JEDA ANTAR TEMBAKAN
    }
}
```

DINACOM.DINUS.AC.ID



dinacom@dinus.id



fb.com/dinacomOfficial



[@dinacomOfficial](https://twitter.com/dinacomOfficial)



Lanjutan dari main.js (function update)

```
this.player.body.velocity.x = 0; //KECEPATAN PLAYER X 0
this.bos.body.velocity.x = 0; // KEVEPATAN BOS X 0
if (this.cursors.left.isDown){ //JIKA KEYBOARD KIRI KEPENCET
    this.player.body.velocity.x = -150;
    if (this.facing !== 'left'){ //JIKA POSISI PLAYER TIDAK MENENGOK KIRI
        //PLAYER DI SET NENGOK KE KIRI DAN MENJALANKAN ANIMASI JALAN KE KIRI
        this.player.animations.play('left');
        this.facing = 'left';
    }
} else if (this.cursors.right.isDown){
    this.player.body.velocity.x = 150;
    if (this.facing !== 'right'){
        this.player.animations.play('right');
        this.facing = 'right';
    }
} else {
    if (this.facing !== 'idle'){
        this.player.animations.stop();
        if (this.facing === 'left'){
            this.player.frame = 0;
        } else {
            this.player.frame = 5;
        }
    }
    this.facing = 'idle';
}

if(this.game.time.now > this.bosjump){
    this.bos.body.velocity.y = -250;
    this.bosjump = this.game.time.now + 1800;
}

if (this.jumpButton.isDown && this.player.body.onFloor() && this.game.time.now > this.jumpTimer)
{
    this.player.body.velocity.y = -250;
    this.jumpTimer = this.game.time.now + 750;
}
},
```

DINACOM.DINUS.AC.ID



dinacom@dinus.id



[fb.com/dinacomOfficial](https://www.facebook.com/dinacomOfficial)



[@dinacomOfficial](https://twitter.com/dinacomOfficial)



Lanjutan dari main.js (function render, cek, finish, tambah)

```
render:function(){
  //this.game.debug.bodyInfo(this.player, 16, 24);
},
cek:function(a,b){
  if(a.body.blocked.right){
    a.body.velocity.x=-200;
  } else if(a.body.blocked.left) {
    a.body.velocity.x=200;
  } else if (a.body.touching.right) {
    a.body.velocity.x=-400;
  } else if (a.body.touching.left) {
    a.body.velocity.x=400;
  }
},
finish:function() {
  this.score+=50;
  if (this.game.device.localStorage) {
    localStorage.score = this.score;
  }
  this.game.state.start('finish');
},
tambah:function(obj1,obj2){

  this.aduar.play();
  if(obj2 == this.bigstar1){
    this.score+=30;
  } else {
    this.score+=10;
  }
  obj2.kill();
},
gameover:function(){
  if (this.game.device.localStorage) {
    localStorage.score = this.score;
  }
  this.game.state.start('gameover');
}
}
```

Diatas telah dibahas bagaimana kita membangun state main atau inti dari game ini, mulai dari kondisi player, musuh, coin, peletakan dan kondisi-kondisi lain nya. Didalam state **main** kita juga telah membuat beberapa fungsi tambahan untuk mendukung aktifitas game seperti pada umumnya. Fungsi-fungsi tambahan seperti **tambah** untuk mengupdate score apabila kita menabrak atau mengambil coin, fungsi **finish** apabila kita telah menyelesaikan maps pada game ini, fungsi **gameover** apabila kita mati terkena serangan musuh atau jatuh ke dalam jurang, dan beberapa fungsi tambahan lainnya. Selanjutnya kita akan membuat state **gameover** dan state **finish**

DINACOM.DINUS.AC.ID



dinacom@dinus.id



fb.com/dinacomOfficial



[@dinacomOfficial](https://twitter.com/dinacomOfficial)



GameOver.js(assets/js/gameover.js)

```
var gameover = function(game){
    console.log("%cGameOver", "color:white; background:Yellow");
}

gameover.prototype = {
    create: function(){
        this.over = this.game.add.audio('audio_over');
        this.over.play();

        // BACKGROUND
        this.background = this.game.add.tileSprite(0, 0, 1365, 650, "background");
        this.background.autoScroll(-90,0);

        this.over = this.game.add.text(this.game.width/2, 100, 'YOU FAILED FINISH THE
MAP!!', { fill: '#ffffff' });
        this.over.anchor.setTo(0.5,0.5);
        this.game.add.tween(this.over).to({y:150}, 1000, Phaser.Easing.Linear, true, 0, 100, true);

        var tombol =
this.game.add.button(this.game.width/2,200,"img_tombolmainlagi",this.mainlagi,this);
        tombol.anchor.setTo(0.5,0.5);
        this.SCOREAkhir = this.game.add.text(this.game.width/2, 260, 'Score Anda : '+
localStorage.score, { fill: '#ffffff' });
        this.SCOREAkhir.anchor.setTo(0.5,0.5);
        this.SCOREAkhir = this.game.add.text(this.game.width/2, 350, 'High Score : '+
localStorage.highScore + ' ( '+localStorage.nama+' )', { fill: '#ffffff' });
        this.SCOREAkhir.anchor.setTo(0.5,0.5);

    },
    mainlagi: function(){
        this.game.state.start("main");
    }
}
```

Pada state **gameover** terdapat button main lagi apabila di klik akan memanggil fungsi main lagi didalam fungsi mainlagi memanggil dan memulai atau meload state main, jadi kita akan kembali kedalam state main untuk memulai game lagi. Pada state ini juga akan ditampilkan highscore dan score yang kita dapat didalam game.

DINACOM.DINUS.AC.ID



dinacom@dinus.id



fb.com/dinacomOfficial



[@dinacomOfficial](https://twitter.com/dinacomOfficial)



finish.js(assets/js/finish.js)

```
var finish = function(game){
    console.log("%cFinish", "color:white; background:green");
}

finish.prototype = {
    create: function() {
        this.over = this.game.add.audio('audio_over');
        this.over.play();

        // BACKGROUND
        this.background = this.game.add.tileSprite(0, 0, 1365, 650, "background");
        this.background.autoScroll(-90,0);

        this.over = this.game.add.text(this.game.width/2, 100, 'CONGRATULATIONS, YOU
FINISHED THE MAP!!!', { font:'20px Arial',fill: '#fff' });
        this.over.anchor.setTo(0.5,0.5);
        this.game.add.tween(this.over).to({y:150}, 1000, Phaser.Easing.Linear, true, 0, 100, true);

        var tombol =
this.game.add.button(this.game.width/2,400,"img_tombolmainlagi",this.mainlagi,this);
        tombol.anchor.setTo(0.5,0.5);
        if(localStorage.score>localStorage.highScore) {
            this.ghg = this.game.add.text(this.game.width/2, 250, 'CONGRATULATIONS, YOU
GOT THE HIGH SCORE', { font:'46px Arial',fill: '#111' });
            this.ghg.anchor.setTo(0.5,0.5);
            localStorage.highScore = localStorage.score;
            localStorage.nama = prompt("Masukan Nama Anda : ");
        }

        this.SCOREAkhir = this.game.add.text(this.game.width/2, 350, 'Score Anda : '+
localStorage.score, { fill: '#ffffff' });
        this.SCOREAkhir.anchor.setTo(0.5,0.5);
        this.SCOREAkhir = this.game.add.text(this.game.width/2, 475, 'High Score : '+ localStorage.highScore + ' (
'+localStorage.nama+')', { fill: '#ffffff' });
        this.SCOREAkhir.anchor.setTo(0.5,0.5);

    },
    mainlagi: function(){
        this.game.state.start("main");
    }
}
```

Pada state **finish** terdapat button main lagi apabila di klik akan memanggil fungsi main lagi didalam fungsi mainlagi memanggil dan memulai atau meload state main, jadi kita akan kembali kedalam state main untuk memulai game lagi. Seperti pada state **gameover** kita juga akan menampilkan highscore sebelumnya dan score yang kita dapatkan dalam game.

DINACOM.DINUS.AC.ID



dinacom@dinus.id



[fb.com/dinacomOfficial](https://www.facebook.com/dinacomOfficial)



[@dinacomOfficial](https://twitter.com/dinacomOfficial)