

## Percakapan:

**“Belajar Bikin Game Bird Obstacle dari Nol Pakai HTML, CSS & JavaScript – Tutorial Lengkap untuk Pemula”**

### Karakter:

- **Kak Sulthon** (guru santai, jelas, dan sabar)
  - **Adik Reza** (murid pemula, excited, banyak tanya)
- 

**Kak Sulthon:** Halo semuanya! Selamat datang di tutorial seru hari ini. Kita mau bikin game klasik yang bikin ketagihan, yaitu **Bird Obstacle!** Burung kecil yang harus menghindari rintangan pipa yang terus datang. Siapa yang pernah main game sejenis sampai jari pegel? ☺♂☺

**Adik Reza:** Aku Kak! Main cuma beberapa detik langsung nabrak pipa terus. Susah tapi nagih banget. Bisa bikin sendiri gak Kak?

**Kak Sulthon:** Bisa dong! Kita pakai cuma **HTML + CSS + JavaScript** murni, tanpa library atau framework ribet. Cocok buat pemula sampai yang sudah agak jago. Hari ini kita akan pelajari:

1. Struktur HTML dasar untuk game
2. CSS biar tampilan kerenn dan responsif
3. JavaScript untuk gravitasi, lompat, pipa random, deteksi tabrakan, skor, dan game over
4. Animasi sayap burung sederhana
5. Tips supaya game kamu makin pro setelah selesai

Siap mulai? Yuk langsung praktek!

**Adik Reza:** Siap Kak! Folder-nya gimana dulu?

**Kak Sulthon:** Bagus, langsung action. Buat folder baru, kasih nama misalnya `bird-obstacle-sulthon`. Di dalam folder itu buat struktur seperti ini:

text

```
bird-obstacle-sulthon/
├── index.html
├── style.css
└── script.js
└── images/
    ├── Bird.png      ← burung sayap normal
    └── Bird-2.png    ← burung sayap terangkat (untuk animasi)
```

└─ background-img.png ← background langit atau apa saja yang kamu suka

**Adik Reza:** Kenapa burungnya butuh dua gambar Kak?

**Kak Sulthon:** Biar kelihatan flap-flap sayapnya pas lompat. Nanti kita ganti gambarnya cepat-cepat supaya animasinya hidup.

Oke, kita mulai dari **index.html** dulu ya.

HTML

```
<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
    <title>Bird Obstacle - Game Sulthon</title>
    <link rel="stylesheet" href="style.css" />
</head>
<body>
    <div class="background"></div>

    <!-- Burung utama -->
    

    <!-- Pesan awal / game over -->
    <div class="message">
        TEKAN ENTER UNTUK MEMULAI<br>
        Gunakan <span style="color:lime">↑</span> atau <b>SPASI</b> untuk
        terbang!
    </div>

    <!-- Tampilan skor -->
    <div class="score">
        <span class="score_title">Score :</span>
        <span class="score_val">0</span>
    </div>

    <script src="script.js" defer></script>
```

```
</body>  
</html>
```

**Adik Reza:** Background-nya gimana biar bagus Kak?

**Kak Sulthon:** Itu tugas **style.css**! Kita bikin background full screen, burung di posisi awal, pipa dengan efek gradient keren, dan skor yang mencolok.

Berikut contoh **style.css** lengkap:

CSS

```
* {  
    margin: 0;  
    padding: 0;  
    box-sizing: border-box;  
}  
  
.background {  
    position: fixed;  
    inset: 0;  
    background: url('images/background-img.png') no-repeat center center  
fixed;  
    background-size: cover;  
    z-index: -1;  
}  
  
.bird {  
    position: fixed;  
    width: 60px;  
    height: 45px;  
    left: 25vw;  
    top: 40vh;  
    z-index: 100;  
    transition: transform 0.08s;  
}  
  
.pipe_sprite {  
    position: fixed;
```

```
width: 70px;
background: linear-gradient(to bottom, #0130a1, #021161);
border: 6px solid #008ae6;
border-radius: 30px;
z-index: 50;
}

.message {
position: absolute;
top: 50%;
left: 50%;
transform: translate(-50%, -50%);
color: #ff0000;
font-size: 3.5rem;
font-weight: bold;
text-align: center;
text-shadow: 3px 3px 8px black;
background: rgba(0,0,0,0.65);
padding: 40px 60px;
border-radius: 20px;
z-index: 200;
}

.score {
position: fixed;
top: 20px;
left: 30px;
color: white;
font-size: 4.5rem;
font-weight: bold;
-webkit-text-stroke: 2px #d60707;
z-index: 150;
}

.score_val {
color: #0de9b2;
```

```

}

/* Responsif untuk HP/tablet */

@media (max-width: 768px) {
    .pipe_sprite { width: 14vw; }
    .bird { width: 90px; height: 70px; }
    .message { font-size: 2.2rem; padding: 25px 40px; }
    .score { font-size: 3rem; left: 15px; }
}

```

**Adik Reza:** Keren Kak! Sekarang bagian JavaScript-nya dong, yang bikin game bergerak.

**Kak Sulthon:** Ini bagian paling seru! Kita mulai dari variabel dasar sampai fungsi utama.

### JavaScript

```

// Variabel dasar game

let move_speed = 3.8;
let gravity = 0.5;
let game_state = 'Start';

const bird = document.querySelector('.bird');
const birdImg = document.getElementById('bird-1');
const score_val = document.querySelector('.score_val');
const message = document.querySelector('.message');

let bird_dy = 0; // kecepatan vertikal burung
let score = 0;

// Mulai game saat tekan Enter
document.addEventListener('keydown', (e) => {
    if (e.key === 'Enter' && game_state !== 'Play') {
        game_state = 'Play';
        message.innerHTML = '';
        score = 0;
        score_val.innerHTML = '0';
        bird.style.top = '40vh';
        bird_dy = 0;
    }
})

```

```

// Hapus semua pipa lama
document.querySelectorAll('.pipe_sprite').forEach(p => p.remove());

playGame();
}

}) ;

function playGame() {
// 1. Fungsi gerakkan pipa & cek tabrakan
function move() {
if (game_state !== 'Play') return;

document.querySelectorAll('.pipe_sprite').forEach(pipe => {
let pipeRect = pipe.getBoundingClientRect();
let birdRect = bird.getBoundingClientRect();

if (pipeRect.right < 0) {
pipe.remove();
return;
}

// Deteksi tabrakan
if (
birdRect.left < pipeRect.right &&
birdRect.right > pipeRect.left &&
birdRect.top < pipeRect.bottom &&
birdRect.bottom > pipeRect.top
) {
game_state = 'End';
message.innerHTML = `GAME OVER!<br>Skor akhir:
${score}<br><br>Tekan ENTER untuk main lagi`;
return;
}

// Tambah skor saat berhasil lewati pipa
}

```

```

        if (pipe.increase_score === '1' && pipeRect.right <
birdRect.left) {
            score++;
            score_val.innerHTML = score;
            pipe.increase_score = '0'; // cegah nambah berulang
        }

        pipe.style.left = (pipeRect.left - move_speed) + 'px';
    });

    requestAnimationFrame(move);
}

// 2. Gravitasi & lompat
function apply_gravity() {
    if (game_state !== 'Play') return;

    bird_dy += gravity;
    let currentTop = parseFloat(bird.style.top) || (40 *
window.innerHeight / 100);
    bird.style.top = (currentTop + bird_dy) + 'px';

    let birdRect = bird.getBoundingClientRect();
    if (birdRect.top <= 0 || birdRect.bottom >= window.innerHeight) {
        game_state = 'End';
        message.innerHTML = `GAME OVER!<br>Nabrak ${birdRect.top <= 0 ?
'plafon' : 'lantai'}!<br>Skor: ${score}`;
    }

    requestAnimationFrame(apply_gravity);
}

// 3. Kontrol lompat (↑ atau Spasi)
function flap(e) {
    if (game_state !== 'Play') return;
    if (e.key === 'ArrowUp' || e.key === ' ') {

```

```

        bird_dy = -9.5; // kekuatan lompat (negatif = ke atas)
        birdImg.src = 'images/Bird-2.png'; // sayap naik
        setTimeout(() => birdImg.src = 'images/Bird.png', 120); //
    kembali normal
    }
}

document.addEventListener('keydown', flap);

// 4. Buat pipa baru secara periodik
let pipe_timer = 0;
const PIPE_GAP = 180; // jarak vertikal antar pipa (pixel)

function create_pipe() {
    if (game_state !== 'Play') return;

    pipe_timer++;
    if (pipe_timer > 85) { // muncul setiap ~1.4 detik (sesuaikan sesuai
kecepatan)
        pipe_timer = 0;

        const holePos = Math.floor(Math.random() * 320) + 120; // posisi
lubang random

        // Pipa atas
        let pipeTop = document.createElement('div');
        pipeTop.className = 'pipe_sprite';
        pipeTop.style.height = holePos + 'px';
        pipeTop.style.top = '0px';
        pipeTop.style.left = '100vw';

        // Pipa bawah
        let pipeBottom = document.createElement('div');
        pipeBottom.className = 'pipe_sprite';
        pipeBottom.style.height = (window.innerHeight - holePos -
PIPE_GAP) + 'px';
        pipeBottom.style.top = (holePos + PIPE_GAP) + 'px';
    }
}

```

```

    pipeBottom.style.left = '100vw';
    pipeBottom.increase_score = '1';

    document.body.appendChild(pipeTop);
    document.body.appendChild(pipeBottom);

}

requestAnimationFrame(create_pipe);
}

// Jalankan semua loop animasi
requestAnimationFrame(move);
requestAnimationFrame(apply_gravity);
requestAnimationFrame(create_pipe);
}

```

**Adik Reza:** Panjang banget tapi masuk akal Kak! Aku mulai paham alurnya.

**Kak Sulthon:** Bagus! Setelah ini kamu bisa coba jalankan di browser. Tekan Enter → lompat pakai ↑ atau Spasi → hindari pipa → kumpulin skor sebanyak mungkin.

Beberapa ide upgrade biar game **Bird Obstacle** kamu makin keren:

- Tambah suara (point, mati, flap sayap)
- Simpan high score pakai localStorage
- Tambah level (pipa muncul lebih cepat seiring skor naik)
- Tampilan mobile (pakai touch event untuk lompat)
- Efek partikel saat nabrak
- Tambah start screen + instruksi lebih detail

**Adik Reza:** Makasih banyak Kak Sulthon! Tutorialnya jelas banget, langsung pengen coba bikin dan modifikasi sendiri. Nanti kalau sudah jadi aku share ya!

**Kak Sulthon:** Sama-sama Reza! Semangat coding-nya. Kalau ada bug atau mau improve bagian tertentu, tinggal tanya lagi. Selamat bikin **Bird Obstacle** versi terbaikmu! ☺☺

**SAYA TAMBAH SEDIKIT CARI EFEK SUARA/SOUND ,CARI GAMBANG BRUNG NYA MINIMAL 3 GAMBAR DAN GAMBAR ITU HARUS PNG BUKAN JPG**

# **SELAMAT BELAJAR DAN SUKSES TERUS**