<!DOCTYPE html>

<html lang="id">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Burger Mania 2024</title>

<script src="https://cdn.tailwindcss.com"></script>

<style>

@import url('https://fonts.googleapis.com/css2?family=Fredoka+One&family=Inter:wght@400;700&display=swap');

body {

font-family: 'Inter', sans-serif;

touch-action: manipulation; /\* Mencegah zoom saat double tap di mobile \*/

}

.title-font {

font-family: 'Fredoka One', cursive;

}

/\* Animasi untuk pelanggan datang dan pergi \*/

.customer-enter {

animation: slideIn 0.5s forwards;

}

.customer-exit {

animation: slideOut 0.5s forwards;

}

@keyframes slideIn {

from { transform: translateX(-100%); opacity: 0; }

to { transform: translateX(0); opacity: 1; }

}

@keyframes slideOut {

from { transform: translateX(0); opacity: 1; }

to { transform: translateX(100%); opacity: 0; }

}

/\* Animasi untuk bubble pesanan \*/

.bubble-show {

animation: popIn 0.3s cubic-bezier(0.68, -0.55, 0.27, 1.55) forwards;

}

@keyframes popIn {

from { transform: scale(0.5); opacity: 0;}

to { transform: scale(1); opacity: 1;}

}

/\* Style untuk bahan-bahan burger \*/

.ingredient {

transition: all 0.2s ease-in-out;

}

.ingredient:hover {

transform: scale(1.1);

}

.ingredient:active {

transform: scale(0.95);

}

#plate > div {

transition: transform 0.3s ease, margin-bottom 0.3s ease;

}

/\* Style untuk pesan notifikasi \*/

.message-box {

animation: fadeInOut 2s forwards;

}

@keyframes fadeInOut {

0%, 100% { opacity: 0; transform: translateY(20px); }

10%, 90% { opacity: 1; transform: translateY(0); }

}

/\* Gradient untuk tombol \*/

.btn-gradient {

background-image: linear-gradient(to right, #f87171, #fb923c);

color: white;

transition: transform 0.2s, box-shadow 0.2s;

}

.btn-gradient:hover {

transform: translateY(-2px);

box-shadow: 0 4px 10px rgba(0,0,0,0.2);

}

</style>

</head>

<body class="bg-amber-100 min-h-screen flex items-center justify-center p-2">

<div id="game-container" class="bg-white/70 backdrop-blur-sm rounded-2xl shadow-2xl w-full max-w-4xl p-4 md:p-6 relative overflow-hidden">

<!-- Header Game -->

<div class="text-center mb-4 border-b-4 border-red-400 pb-4">

<h1 class="text-4xl md:text-5xl title-font text-amber-600">Burger Mania</h1>

<p class="text-gray-600">Buat burger sesuai pesanan pelanggan!</p>

</div>

<!-- Panel Status -->

<div class="flex justify-around items-center bg-red-400 text-white p-3 rounded-xl shadow-lg mb-4 text-lg md:text-xl">

<div>

<span>Skor: </span>

<span id="score" class="font-bold">0</span>

</div>

<div>

<span>Waktu: </span>

<span id="timer" class="font-bold">0</span>

</div>

</div>

<!-- Area Game Utama -->

<div class="grid grid-cols-1 lg:grid-cols-2 gap-4 md:gap-6 min-h-[400px]">

<!-- Area Pelanggan -->

<div id="customer-section" class="bg-blue-200/50 rounded-xl p-4 flex flex-col items-center justify-center relative min-h-[250px] lg:min-h-0">

<div id="customer-area" class="relative w-48 h-56 flex items-center justify-center">

<!-- Placeholder untuk saat tidak ada pelanggan -->

<div id="customer-placeholder" class="text-center text-gray-500">

<div class="text-4xl">🍔</div>

<div>Menunggu Pelanggan...</div>

</div>

<!-- Pelanggan dan pesanannya akan muncul di sini -->

</div>

</div>

<!-- Area Memasak -->

<div id="cooking-section" class="bg-amber-200/50 rounded-xl p-4 flex flex-col justify-between">

<!-- Area Rakit Burger -->

<div id="assembly-area" class="w-full h-48 bg-gray-100 rounded-lg shadow-inner flex items-end justify-center p-2 mb-4">

<div id="plate" class="relative w-48 h-full flex flex-col-reverse items-center">

<!-- Bahan burger yang dirakit akan muncul di sini -->

</div>

</div>

<!-- Area Pilihan Bahan -->

<div id="ingredients-area" class="grid grid-cols-3 sm:grid-cols-4 gap-2 text-center">

<!-- Tombol bahan-bahan akan di-generate oleh JS -->

</div>

</div>

</div>

<!-- Tombol Kontrol Game -->

<div id="controls" class="text-center mt-4">

<button id="start-button" class="title-font text-2xl py-3 px-10 rounded-full btn-gradient shadow-lg">Mulai Game</button>

</div>

<!-- Kotak Pesan Notifikasi -->

<div id="message-container" class="absolute inset-0 flex items-center justify-center pointer-events-none"></div>

</div>

<script>

document.addEventListener('DOMContentLoaded', () => {

// --- Konfigurasi Game ---

const INGREDIENTS = {

'roti-bawah': { name: 'Roti Bawah', html: '<div class="w-24 h-4 bg-orange-400 rounded-t-md"></div>' },

'patty': { name: 'Patty', html: '<div class="w-20 h-4 bg-yellow-900 rounded-md"></div>' },

'keju': { name: 'Keju', html: '<div class="w-20 h-2 bg-yellow-400"></div>' },

'selada': { name: 'Selada', html: '<div class="w-24 h-3 bg-green-500 rounded-full"></div>' },

'tomat': { name: 'Tomat', html: '<div class="w-20 h-3 bg-red-500 rounded-full"></div>' },

'roti-atas': { name: 'Roti Atas', html: '<div class="w-24 h-8 bg-orange-400 rounded-t-full relative"><div class="absolute top-2 left-4 w-1 h-1 bg-white rounded-full"></div><div class="absolute top-3 left-8 w-1 h-1 bg-white rounded-full"></div><div class="absolute top-2 left-12 w-1 h-1 bg-white rounded-full"></div><div class="absolute top-4 left-16 w-1 h-1 bg-white rounded-full"></div></div>' },

};

const ORDER\_TIME\_LIMIT = 20; // Waktu per pesanan dalam detik

const CHARACTERS = [

{ emoji: '🧑', color: 'bg-red-300' },

{ emoji: '👱‍♀️', color: 'bg-yellow-300' },

{ emoji: '👨‍🦱', color: 'bg-green-300' },

{ emoji: '👵', color: 'bg-purple-300' },

{ emoji: '👨‍🚀', color: 'bg-gray-300' },

];

// --- Elemen DOM ---

const scoreEl = document.getElementById('score');

const timerEl = document.getElementById('timer');

const customerArea = document.getElementById('customer-area');

const customerPlaceholder = document.getElementById('customer-placeholder');

const plate = document.getElementById('plate');

const ingredientsArea = document.getElementById('ingredients-area');

const startButton = document.getElementById('start-button');

const messageContainer = document.getElementById('message-container');

// --- State Game ---

let score = 0;

let timeLeft = 0;

let currentOrder = [];

let playerAssembly = [];

let gameInterval = null;

let gameActive = false;

// --- Fungsi-Fungsi Game ---

// Fungsi untuk menampilkan pesan notifikasi

function showMessage(text, type = 'success') {

const color = type === 'success' ? 'bg-green-500' : 'bg-red-500';

const messageBox = document.createElement('div');

messageBox.className = `message-box text-white font-bold py-2 px-4 rounded-lg shadow-xl ${color}`;

messageBox.textContent = text;

messageContainer.innerHTML = ''; // Hapus pesan sebelumnya

messageContainer.appendChild(messageBox);

}

// Inisialisasi area bahan-bahan

function setupIngredients() {

ingredientsArea.innerHTML = '';

for (const key in INGREDIENTS) {

const ingredient = INGREDIENTS[key];

const button = document.createElement('button');

button.dataset.ingredient = key;

button.className = 'ingredient p-2 bg-white rounded-lg shadow-md flex flex-col items-center justify-center disabled:opacity-50 disabled:cursor-not-allowed';

button.disabled = true;

// Miniatur bahan

const preview = document.createElement('div');

preview.innerHTML = ingredient.html;

preview.className = 'h-12 w-full flex items-center justify-center';

button.appendChild(preview);

// Nama bahan

const name = document.createElement('span');

name.textContent = ingredient.name;

name.className = 'text-xs text-gray-700 mt-1';

button.appendChild(name);

button.addEventListener('click', () => handleIngredientClick(key));

ingredientsArea.appendChild(button);

}

}

// Memperbarui tampilan tumpukan burger

function updateAssemblyDisplay() {

plate.innerHTML = '';

playerAssembly.forEach((key, index) => {

const ingredientEl = document.createElement('div');

ingredientEl.innerHTML = INGREDIENTS[key].html;

ingredientEl.style.marginBottom = `${-10}px`;

ingredientEl.style.transform = `translateY(${index \* -5}px)`;

plate.appendChild(ingredientEl);

});

}

// Handler saat pemain mengklik bahan

function handleIngredientClick(ingredientKey) {

if (!gameActive) return;

playerAssembly.push(ingredientKey);

updateAssemblyDisplay();

// Cek apakah bahan yang ditambahkan benar

const currentStep = playerAssembly.length - 1;

if (playerAssembly[currentStep] !== currentOrder[currentStep]) {

// Jika salah

showMessage('Bahan Salah!', 'error');

playerAssembly = []; // Reset rakitan

setTimeout(updateAssemblyDisplay, 200);

} else if (playerAssembly.length === currentOrder.length) {

// Jika burger selesai dan benar

clearInterval(gameInterval);

showMessage('Pesanan Selesai!', 'success');

updateScore(10 + timeLeft); // Tambah skor + bonus waktu

setTimeout(startNewRound, 2000); // Lanjut ke pesanan berikutnya

}

}

// Membuat pesanan baru secara acak

function generateOrder() {

const availableIngredients = ['patty', 'keju', 'selada', 'tomat'];

const orderSize = Math.floor(Math.random() \* 3) + 2; // 2-4 bahan isi

let order = ['roti-bawah'];

for (let i = 0; i < orderSize; i++) {

const randomIndex = Math.floor(Math.random() \* availableIngredients.length);

order.push(availableIngredients[randomIndex]);

}

order.push('roti-atas');

return order;

}

// Menampilkan pelanggan dan pesanannya

function displayCustomerAndOrder() {

customerArea.innerHTML = '';

customerPlaceholder.classList.add('hidden');

const customerData = CHARACTERS[Math.floor(Math.random() \* CHARACTERS.length)];

// Div pelanggan

const customerDiv = document.createElement('div');

customerDiv.className = 'w-32 h-32 rounded-full flex items-center justify-center text-6xl shadow-lg customer-enter';

customerDiv.classList.add(customerData.color);

customerDiv.textContent = customerData.emoji;

// Div bubble pesanan

const orderBubble = document.createElement('div');

orderBubble.className = 'absolute -top-8 -right-8 bg-white p-2 rounded-lg shadow-lg flex space-x-1 border-2 border-gray-300 bubble-show';

currentOrder.forEach(key => {

const icon = document.createElement('div');

icon.className = 'w-6 h-6 flex items-center justify-center scale-75';

icon.innerHTML = INGREDIENTS[key].html;

orderBubble.appendChild(icon);

});

customerArea.appendChild(customerDiv);

customerArea.appendChild(orderBubble);

}

// Memperbarui skor

function updateScore(points) {

score += points;

scoreEl.textContent = score;

}

// Memulai timer

function startTimer() {

timeLeft = ORDER\_TIME\_LIMIT;

timerEl.textContent = timeLeft;

gameInterval = setInterval(() => {

timeLeft--;

timerEl.textContent = timeLeft;

if (timeLeft <= 0) {

clearInterval(gameInterval);

if(gameActive){

showMessage('Waktu Habis!', 'error');

setTimeout(startNewRound, 2000);

}

}

}, 1000);

}

// Memulai ronde/pesanan baru

function startNewRound() {

if (!gameActive) return;

// Animasi pelanggan pergi

const currentCustomer = customerArea.querySelector('.customer-enter');

if(currentCustomer) {

currentCustomer.classList.remove('customer-enter');

currentCustomer.classList.add('customer-exit');

}

// Reset state

playerAssembly = [];

currentOrder = generateOrder();

updateAssemblyDisplay();

clearInterval(gameInterval);

// Setelah animasi selesai, munculkan pelanggan baru

setTimeout(() => {

displayCustomerAndOrder();

startTimer();

}, 500);

}

// Memulai atau mengakhiri permainan

function toggleGameState() {

gameActive = !gameActive;

if (gameActive) {

// Mulai Game

startButton.textContent = 'Berhenti';

startButton.classList.remove('btn-gradient');

startButton.classList.add('bg-gray-400', 'hover:bg-gray-500');

score = 0;

scoreEl.textContent = score;

document.querySelectorAll('#ingredients-area button').forEach(b => b.disabled = false);

startNewRound();

} else {

// Hentikan Game

startButton.textContent = 'Mulai Game';

startButton.classList.add('btn-gradient');

startButton.classList.remove('bg-gray-400', 'hover:bg-gray-500');

clearInterval(gameInterval);

timerEl.textContent = '0';

customerArea.innerHTML = '';

customerPlaceholder.classList.remove('hidden');

plate.innerHTML = '';

showMessage(`Game Selesai! Skor Akhir: ${score}`, 'success');

document.querySelectorAll('#ingredients-area button').forEach(b => b.disabled = true);

}

}

// Inisialisasi awal

setupIngredients();

startButton.addEventListener('click', toggleGameState);

});

</script>

</body>

</html>