

LAPORAN PERKEMBANGAN WEBISTE UGC

(*Undiknas Gaming Center*)

Dhais Afla Rysy (42330030)

Ida Bagus Anom Puja Satyabratha (42330036)

Dhegas Alkadico Susilo (42330040)

1. Latar Belakang

Undiknas Gaming Center adalah Website yang berbasis *E-commerce* berfokus menjual produk *video game* dengan menggunakan *Cloud Service* dimana player bisa memainkan game yang mereka beli langsung dari websitenya. proyek lanjutan dari Undiknas hero proyek uts kita, Kondisi kamu mulai proyek ini bahwa UI/UX sudah di develop namun tidak maksimal, Front end pun juga sudah di develop. Fitur saat uts yaitu kami bisa menggunakan parameter untuk menambahkan sebuah game kedalam keranjang itu akan mengupdate semua tergantung dari game yang ditambahkan namun data nya masih fiktif tidak ada data valid nya di project kami kali ini tujuan utama nya adalah untuk mengintegrasikan database dan membaut data valid dari semua game yang kami sediakan. Kami juga akan berusaha membaut API Login/Registrasi, API searh dan Kategori. Sekaligus juga mengembangkan UI/UX yang lebih menarik dan interaktif.

2. Perkembangan Fitur

Sebelum perubahan arsitektur sistem ini, website kamu menggunakan file HTML statis untuk menampilkan konten. File HTML hanya menyajikan konten statis yang tidak bisa berinteraksi dengan pengguna atau database. Artinya, untuk setiap perubahan pada konten, halaman harus diedit secara manual, dan tidak ada interaktivitas dengan server untuk melakukan hal-hal seperti memproses data formulir atau memperbarui tampilan halaman berdasarkan input pengguna.

Namun, setelah mempertimbangkan kebutuhan untuk membuat website lebih dinamis dan berinteraksi dengan database (seperti memuat data game dari database atau memperbarui stok), kita memutuskan untuk mengganti file HTML menjadi PHP. PHP memungkinkan kita untuk mengakses database, memproses data dari pengguna, dan menampilkan konten secara dinamis tanpa perlu memperbarui file HTML secara manual setiap kali ada perubahan. Implementasi dan Perkembangan ini Diserahkan ke Programmer Anom Dan Dhegas

2.2 Langkah-langkah Pergantian dari HTML ke PHP

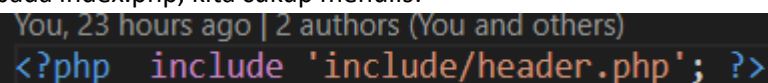
2.2.1 Mengubah Struktur Halaman

Sebelum pergantian ini, website menggunakan 5 halaman HTML terpisah untuk setiap fitur atau tampilan. Masing-masing halaman ini mungkin berisi konten yang bersifat statis dan tidak dapat disesuaikan berdasarkan permintaan pengguna. Ketika menggunakan PHP, kita mulai untuk mengganti file HTML tersebut menjadi file PHP yang lebih dinamis. Berikut adalah langkah-langkah yang diambil untuk mengubah arsitektur sistem:

1. Mengganti Ekstensi File Setiap file HTML yang ada (misalnya index.html, trending.html, checkout.html, dll.) diubah menjadi file PHP dengan ekstensi .php. Sebagai contoh, index.html akan diganti menjadi index.php.
2. Integrasi dengan PHP untuk Mengakses Database Pada PHP, kita dapat menggunakan kode untuk mengakses database dan menampilkan data secara dinamis. Sebagai contoh:
 - Pada halaman index.php, kita bisa mengakses database dan menampilkan daftar game yang tersedia.
 - Halaman trending.php bisa menampilkan game-game yang sedang populer berdasarkan data yang diambil dari database.

Dengan mengganti file HTML menjadi PHP, kita menambahkan kemampuan untuk menampilkan konten berdasarkan data yang ada di database, seperti menampilkan game dengan harga, gambar, dan deskripsi yang diambil dari tabel games dalam database.

3. Menambahkan Header dan Footer Dinamis Alih-alih menyalin dan menempelkan kode header dan footer pada setiap halaman HTML, kita bisa menggunakan fitur PHP untuk menyertakan file header dan footer secara dinamis di seluruh halaman. Sebagai contoh, pada index.php, kita cukup menulis:



```
You, 23 hours ago | 2 authors (You and others)
<?php include 'include/header.php'; ?>
```

Ini akan menyertakan file header.php ke dalam halaman, yang berisi struktur HTML umum seperti navigasi dan metadata, dan memudahkan pengelolaan tampilan secara keseluruhan. Begitu juga dengan footer, kita dapat menyertakan file footer menggunakan kode PHP yang sama.

4. Penggunaan PHP untuk Menangani Formulir dan Input Pengguna Sebelumnya, input dari pengguna (seperti ketika menambahkannya ke keranjang atau checkout) hanya bisa diproses dengan JavaScript atau tidak diproses sama sekali di halaman HTML statis. Dengan PHP, kita dapat menangani form input secara langsung:
 - Misalnya, ketika pengguna menambahkan game ke keranjang belanja, kita dapat memproses data tersebut di server dan memperbarui database.
 - Pada halaman checkout.php, formulir pengisian informasi pengguna dapat diproses menggunakan PHP untuk memastikan bahwa data yang dimasukkan valid dan kemudian diproses untuk melakukan transaksi.

2.2.2 Modifikasi Halaman untuk Interaksi dengan Database

Setiap halaman yang sebelumnya bersifat statis kini diubah menjadi halaman dinamis dengan mengakses database melalui PHP. Berikut adalah beberapa hal yang diperbarui:

1. Halaman Index (index.php)
 - Sebelumnya, daftar game dan kontennya mungkin hardcoded di file HTML.
 - Sekarang, menggunakan PHP dan query SQL untuk mengambil data dari tabel games dan menampilkannya di halaman.

- Contoh kode:

```
<section class="game-grid">
  <?php
  if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
      echo '<div class="game-card">';
      echo ''; // Menampilkan gambar
      echo '<h2>' . $row['title'] . '</h2>';
      echo '<p>Special offer price: $' . number_format(num: $row['price'], decimals: 2) . '</p>';
      echo '<p>Hurry up! This offer expires in:</p>';
      echo '<div class="countdown" id="countdown' . $row['id'] . '">Loading...</div>';
      echo '<button class="add-to-cart" data-title="' . $row['title'] . '" data-price="' . $row['price'] . '" data-id="' . $row['id'] . '">Add to Cart';
      echo '</div>';
    }
  } else {
    echo "No special offers available at the moment.";
  }
  >
</section>
```

2. halaman Trending (trending.php)

- Data yang ditampilkan sekarang diambil dari database menggunakan PHP untuk menampilkan game yang sedang trending.
- Kita bisa memfilter atau mengurutkan data game berdasarkan kategori atau rating dari database, kemudian menampilkannya di halaman.

3. Halaman Checkout (checkout.php)

- Pada halaman ini, PHP digunakan untuk memproses data checkout yang dimasukkan pengguna.
- PHP juga digunakan untuk mengurangi stok game setelah transaksi dilakukan, dengan memperbarui data di database.

4. Halaman Cart (cart.php)

- Cart sebelumnya mungkin dikelola menggunakan JavaScript di sisi klien. Kini, kita menggunakan PHP untuk memproses cart di sisi server dan memastikan bahwa data tetap konsisten antara sesi pengguna dan database.
- PHP menangani data yang ditambahkan ke keranjang dan memastikan perubahan stok tercatat dengan benar di database.

2.3 Manfaat dari Pergantian ke PHP

1. Interaktivitas dan Dinamisme PHP memungkinkan interaktivitas yang lebih tinggi pada website karena dapat memproses data dari pengguna secara dinamis. Misalnya, saat menambahkannya ke keranjang belanja, PHP akan memeriksa ketersediaan stok di database dan mengurangi stok yang sesuai.
2. Manajemen Konten yang Lebih Mudah Dengan menggunakan PHP, kita bisa membuat website yang lebih mudah dikelola, karena konten dapat diambil dan diperbarui langsung dari database tanpa perlu mengedit file HTML setiap kali ada perubahan.
3. Peningkatan Keamanan Menggunakan PHP untuk menangani input pengguna memungkinkan implementasi teknik keamanan yang lebih baik, seperti sanitasi input dan validasi data, untuk mencegah potensi ancaman seperti SQL injection.
4. Pengelolaan Data yang Lebih Terpusat Dengan mengganti file HTML menjadi PHP, data yang terkait dengan game, pengguna, dan transaksi dapat dikelola dalam satu sistem yang terintegrasi, sehingga meningkatkan efisiensi dan keakuratan pengolahan data.

2.4 Kesimpulan

Dengan mengganti file HTML menjadi PHP, website kini dapat berinteraksi lebih baik dengan pengguna, database, dan server. Proses ini meningkatkan fleksibilitas, memungkinkan website untuk menampilkan data dinamis, dan memperkenalkan fungsionalitas baru seperti pengelolaan stok secara otomatis dan checkout yang terintegrasi dengan database.

3. Proses Integrasi Database

pengembangan website selanjutnya adalah pembuatan database yang akan digunakan untuk menyimpan data game yang akan ditampilkan di website. Database ini juga akan mencakup informasi penting seperti nama game, harga, stok, gambar, deskripsi, dan kategori.

1. Pembuatan Database

- Saya mulai dengan membuat database menggunakan MySQL yang dinamakan game_store (atau nama lain sesuai kebutuhan).
- Di dalam database tersebut, saya membuat beberapa tabel yang penting, seperti tabel games yang berfungsi untuk menyimpan data terkait setiap game yang dijual di website.

Jadi query nya seperti ini:

```
CREATE TABLE games (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    title VARCHAR(255) NOT NULL,  
    price DECIMAL(10, 2) NOT NULL,  
    image_url VARCHAR(255),  
    description TEXT,  
    stock INT DEFAULT 0,  
    release_date DATE  
);
```

Tabel games memiliki beberapa kolom penting, yaitu:

- id: Sebagai ID unik untuk setiap game.
- title: Nama game.
- price: Harga game.
- image_url: URL gambar atau poster game.
- description: Deskripsi singkat tentang game.
- stock: Menyimpan jumlah stok yang tersedia.
- release_date: Tanggal rilis game.

2. Memasukkan Data ke dalam Tabel Setelah struktur tabel selesai dibuat, saya kemudian mengisi tabel games dengan data game yang ingin dijual di website, seperti judul, harga, gambar, dan stok.

Contoh Data yang Dimasukkan:

```
INSERT INTO games (title, price, image_url, description, stock, release_date)
```

```
VALUES
```

```
('Until Daylight', 19.99, '/img/UntilDaylight.jpeg', 'An immersive co-op horror game', 100, '2025-12-01'),
```

```
('Elden Ring', 59.99, '/img/EldenRing.jpeg', 'A dark fantasy RPG', 200, '2025-12-01'),
```

```
('Flinthook', 19.99, '/img/Flinthook.jpeg', 'A fast-paced action game', 120, '2025-12-01');
```

3.2 Integrasi dengan PHP (config.php)

Setelah database dan tabel selesai disiapkan, langkah berikutnya adalah menghubungkan website dengan database menggunakan PHP. Hal ini dilakukan melalui file config.php, yang berfungsi untuk mengatur koneksi antara website (PHP) dan database MySQL.

1. Membuat config.php

- File config.php berisi kode untuk menghubungkan website dengan database MySQL. Saya menggunakan fungsi `mysqli_connect` untuk melakukan koneksi ke database.

Contoh kode config.php:

```
1 <?php
2 $servername = "localhost";
3 $username = "root";
4 $password = "";
5 $dbname = "cloud_gaming";
6
7
8 $conn = new mysqli(hostname: $servername, username: $username, password: $password, database: $dbname);
9
10
11 if ($conn->connect_error) {
12     die("Connection failed: " . $conn->connect_error);
13 }
14
15
16 try {
17     $pdo = new PDO(dsn: "mysql:host=$servername;dbname=$dbname", username: $username, password: $password);
18     $pdo->setAttribute(attribute: PDO::ATTR_ERRMODE, value: PDO::ERRMODE_EXCEPTION);
19 } catch (PDOException $e) {
20
21 }
22
```

Dengan file ini, setiap file PHP di website dapat memanfaatkan koneksi ke database dengan menyertakan config.php di bagian awal.

3.3 Mengembangkan Fitur Pengelolaan Stok

Salah satu fitur penting yang perlu diimplementasikan adalah pengelolaan stok. Setiap game yang ada di website harus memiliki stok yang dapat berkurang ketika ada transaksi atau ketika pengguna menambahkannya ke keranjang belanja.

1. Memperbarui Stok Secara Otomatis Untuk memastikan stok game yang tersedia selalu terupdate sesuai dengan transaksi yang dilakukan, saya menambahkan kode untuk mengurangi stok setiap kali game berhasil ditambahkan ke keranjang atau melakukan pembelian.
 - o Sebagai contoh, pada saat pengguna menambahkannya ke keranjang, PHP akan memeriksa jumlah stok yang tersedia melalui query SQL, dan jika stok mencukupi, stok akan dikurangi satu.

Contoh Kode PHP untuk Mengurangi Stok:

```
1 <?php
2 include 'config.php';
3
4 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
5
6     $gameId = isset($_POST['game_id']) ? intval(value: $_POST['game_id']) : 0;
7     $quantity = isset($_POST['quantity']) ? intval(value: $_POST['quantity']) : 1;
8
9
10    if ($gameId <= 0 || $quantity <= 0) {
11        echo json_encode(value: ['success' => false, 'error' => 'Invalid input']);
12        exit;
13    }
14
15
16    $stmt = $conn->prepare(query: "SELECT stock FROM games WHERE id = ?");
17    $stmt->bind_param(types: "i", var: &$gameId);
18    $stmt->execute();
19    $stmt->bind_result(var: &$stock);
20    $stmt->fetch();
21    $stmt->close();
22
23    if ($stock === null) {
24        echo json_encode(value: ['success' => false, 'error' => 'Game not found']);
25        exit;
26    }
27
28    if ($stock >= $quantity) {
29
30        $newStock = $stock - $quantity;
31        $updateStmt = $conn->prepare(query: "UPDATE games SET stock = ? WHERE id = ?");
32        $updateStmt->bind_param(types: "ii", var: &$newStock, vars: &$gameId);
33        $updateStmt->execute();
34        $updateStmt->close();
35    }
```

Pada kode di atas, setelah mengecek jumlah stok, jika stok lebih dari 0, maka stok akan dikurangi satu dan memperbarui data stok di database. Jika stok tidak mencukupi, maka akan muncul pemberitahuan "Stok Habis".

3.4 Uji Coba dan Penyempurnaan

Setelah proses integrasi database dan pengelolaan stok selesai, langkah selanjutnya adalah melakukan uji coba untuk memastikan semua fungsi berjalan dengan baik. Saya menguji berbagai skenario seperti:

- Memeriksa apakah game dapat ditambahkan ke keranjang dengan stok yang cukup.
- Memastikan bahwa stok dikurangi dengan benar setelah transaksi.
- Memastikan bahwa jika stok habis, game tidak dapat ditambahkan ke keranjang.

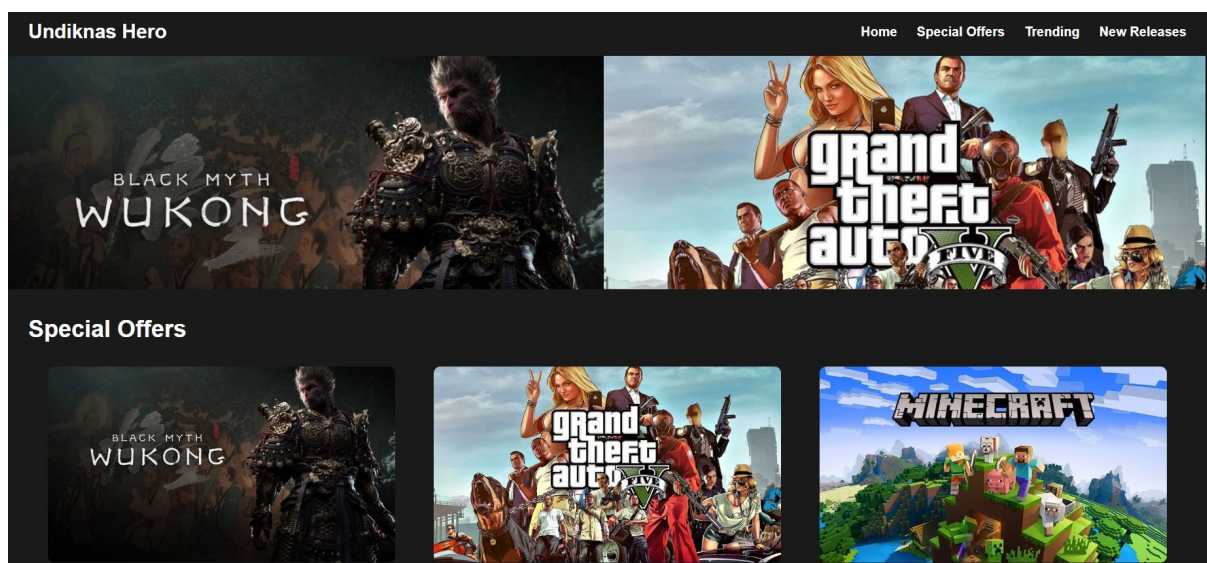
3.5 Kesimpulan

Dengan mengintegrasikan database, website sekarang bisa menampilkan data game yang dinamis dan memperbarui stok secara otomatis berdasarkan interaksi pengguna. Ini memungkinkan website untuk memiliki fitur seperti keranjang belanja yang mengelola stok game secara real-time, memberikan informasi yang lebih akurat kepada pengguna, dan memperbaiki pengalaman berbelanja secara keseluruhan. Database juga membantu menyimpan data secara terpusat, yang membuat pengelolaan data lebih efisien dan aman.

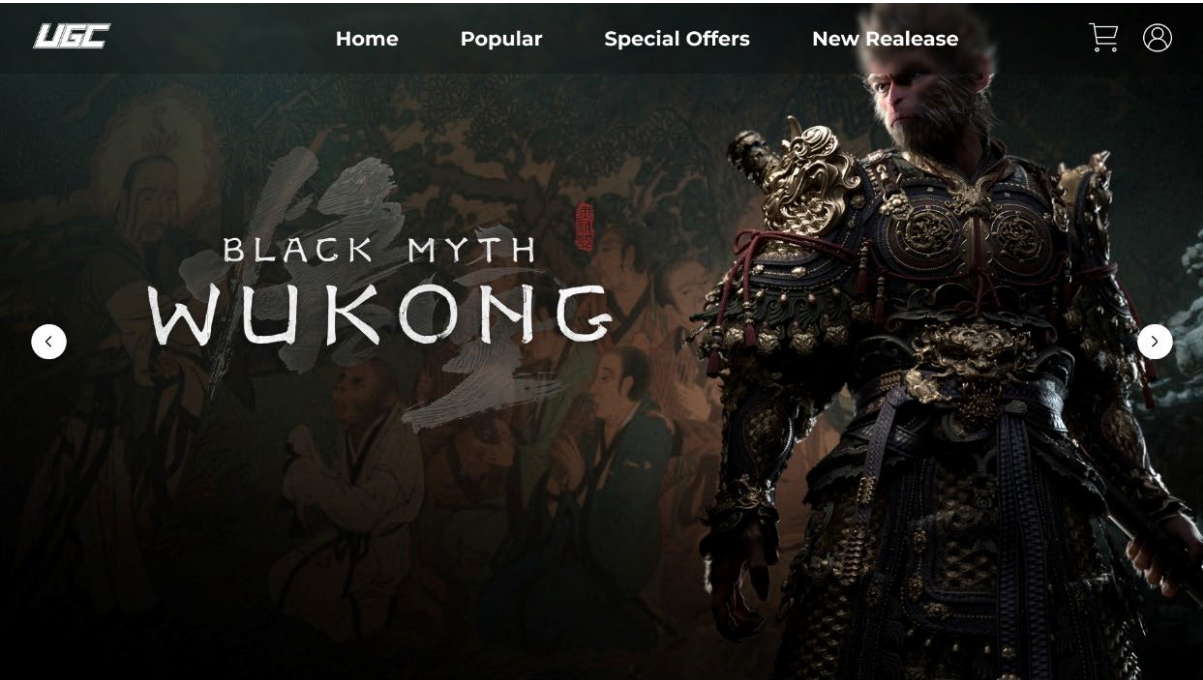
4. Membuat UI/UX baru dan mengimplementasikan Login Page

Setelah Integrasi API Database dan Stock berhasil kami dalam tahap mengembangkan UI ini akan diserahkan oleh UI/UX Specialist Dhaiz. Dengan menggunakan figma kita bisa menentukan keputusan yang tepat dalam pemilihan design.

Ini adalah perbandingan antara yang sebelumnya dan sebelumnya:



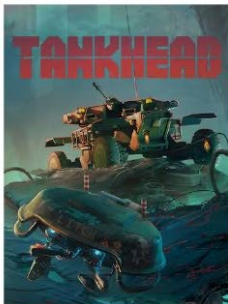
Updated Version (PLAN):



EXPLORE MORE GAMES



Marvel Rivals



TankHead



Infinity Nikki



Final Fantasy VII

View More

5. Login Page

roses pembuatan sistem login dan registrasi merupakan langkah penting dalam pengembangan aplikasi berbasis web untuk memberikan akses dan autentikasi pengguna. Pada proyek ini, sistem login dan registrasi dirancang menggunakan PHP untuk backend dan MySQL sebagai database, serta menerapkan hashing untuk keamanan kata sandi. Tampilan antarmuka (frontend) dibuat menyerupai desain Epic Games, dengan fokus pada estetika dan kemudahan penggunaan.

Pembuatan Database

Tahap awal dimulai dengan pembuatan tabel database untuk menyimpan data pengguna. Tabel diberi nama `user1` dan memiliki kolom-kolom berikut:

- `id` (INT, Primary Key, Auto Increment): Identifikasi unik untuk setiap pengguna.
- `username` (VARCHAR(50), UNIQUE): Nama pengguna untuk login.
- `email` (VARCHAR(100), UNIQUE): Email pengguna.
- `password` (VARCHAR(255)): Kata sandi yang telah di-hash untuk keamanan.
- `created_at` (TIMESTAMP): Waktu pembuatan akun.

Perintah SQL untuk membuat tabel:

```
CREATE TABLE user1 (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  username VARCHAR(50) NOT NULL UNIQUE,  
  email VARCHAR(100) NOT NULL UNIQUE,  
  password VARCHAR(255) NOT NULL,  
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

Pembuatan Halaman Registrasi

Pada tahap ini, halaman registrasi dirancang untuk memungkinkan pengguna mendaftar dengan memasukkan username, email, dan password. Langkah-langkah yang dilakukan adalah:

1. Frontend:
 - Dibuat formulir HTML dengan tiga input: username, email, dan password.
 - Ditambahkan CSS untuk mempercantik antarmuka agar sesuai dengan warna palet mirip Epic Games.
2. Backend:
 - Validasi data input untuk memastikan semua kolom diisi dengan format yang benar.

- Kata sandi di-hash menggunakan password_hash() sebelum disimpan ke database untuk meningkatkan keamanan.
- Data pengguna disimpan di tabel user1 menggunakan perintah SQL INSERT.

Kode PHP utama:

```
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $username = trim($_POST['username']);
    $email = trim($_POST['email']);
    $password = trim($_POST['password']);

    $hashedPassword = password_hash($password, PASSWORD_DEFAULT);

    $query = $conn->prepare("INSERT INTO user1 (username, email, password) VALUES (?, ?, ?);");
    $query->bind_param("sss", $username, $email, $hashedPassword);
    $query->execute();
}
```

Pembuatan Halaman Login

Halaman login memungkinkan pengguna yang telah terdaftar untuk masuk ke sistem dengan username dan password yang valid. Proses pembuatan meliputi:

1. Frontend:
 - Dibuat formulir login dengan input untuk username dan password.
 - Tampilan didesain konsisten dengan halaman registrasi.
2. Backend:
 - Sistem mengambil data pengguna berdasarkan username yang dimasukkan.
 - Kata sandi diverifikasi menggunakan password_verify().
 - Jika berhasil, informasi pengguna disimpan dalam sesi (\$_SESSION) untuk mengelola autentikasi.

Kode PHP utama:

```
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $username = trim($_POST['username']);
    $password = trim($_POST['password']);

    $query = $conn->prepare("SELECT id, password FROM user1 WHERE username = ?");
    $query->bind_param("s", $username);
    $query->execute();
    $query->bind_result($userId, $hashedPassword);
    $query->fetch();

    if ($userId && password_verify($password, $hashedPassword)) {
        $_SESSION['user_id'] = $userId;
        $_SESSION['username'] = $username;
    }
}
```

Keamanan Sistem

Keamanan menjadi prioritas dalam pembuatan sistem login dan registrasi. Beberapa langkah yang diambil:

- Hashing Password: Menggunakan password_hash() untuk mengamankan kata sandi sebelum disimpan.
- Prepared Statements: Semua query SQL menggunakan prepared statements untuk mencegah serangan SQL Injection.
- Validasi Input: Semua data yang dikirimkan pengguna divalidasi untuk mencegah input yang berbahaya.
- Session Management: Sistem login menggunakan \$_SESSION untuk menjaga sesi pengguna yang sedang aktif.

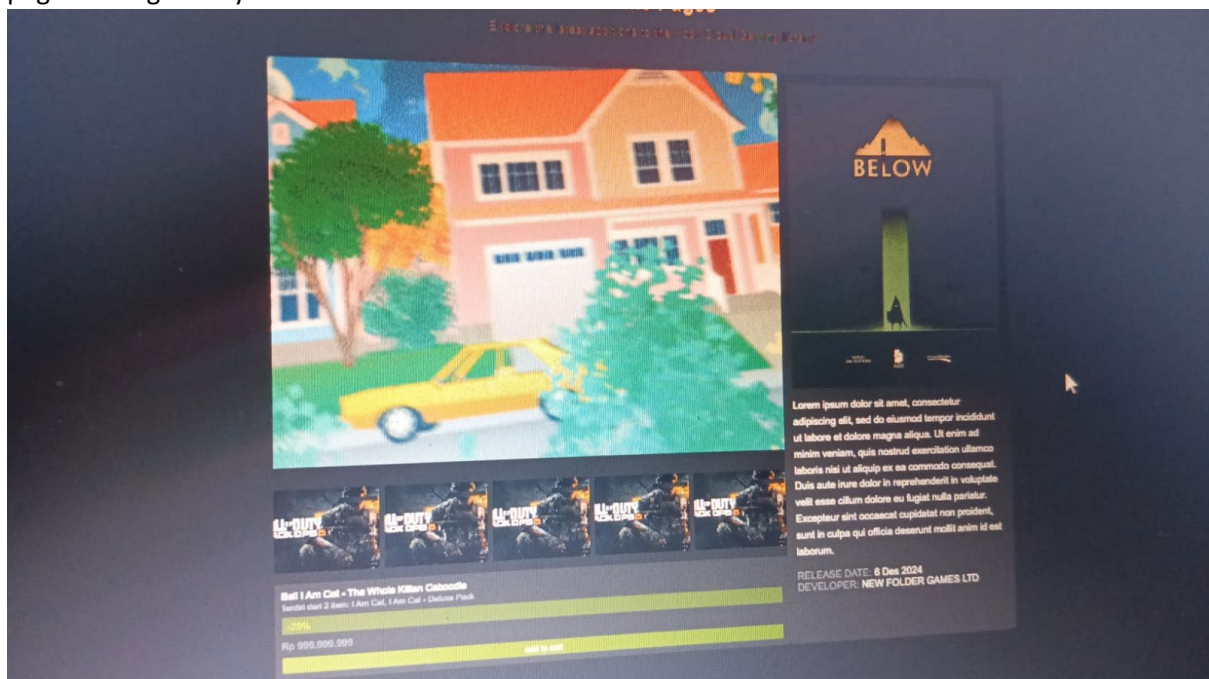
Hasil Akhir

Sistem login dan registrasi telah berhasil diimplementasikan dengan fitur berikut:

- Pengguna dapat mendaftar dengan username, email, dan password yang unik.
- Pengguna yang terdaftar dapat login menggunakan kredensial mereka.
- Sistem mengelola sesi pengguna dengan aman.
- Antarmuka pengguna responsif dan menarik dengan warna palet yang mirip Epic Games.

6. Scrap Idea

Jadi ada beberapa ide yang kita Scrap yang pertama game page dimana setiap game page itu membutuhkan waktu yang sangat banyak untuk dibuat, berikut ada beberapa foto prototipe dari page untuk game nya:



lalu kami juga scarp idea untuk membuat API search dan kategori karena isi dari game kamu juga dikit dan merasa kurang pas dengan fitur tersebut.