

DOKUMENTASI PROYEK: SISTEM REGISTRASI BERBASIS WEB

Deskripsi Proyek

Proyek ini adalah sistem registrasi berbasis web yang terintegrasi dengan API menggunakan Node.js, MySQL sebagai database, dan PHP untuk frontend. Sistem ini memungkinkan pengguna untuk mendaftar paket langganan, mengelola data pengguna, serta menyediakan proses registrasi yang efisien.

INSTALASI DAN SETUP NODE.JS & NPM

1. Instalasi Node.js dan NPM

Windows:

- Download Node.js dari <https://nodejs.org/>
- Jalankan installer dan ikuti instruksi
- Verifikasi instalasi:

```
node -v
```

```
npm -v
```

Linux (Debian/Ubuntu):

```
sudo apt update
```

```
sudo apt install nodejs npm
```

```
node -v
```

```
npm -v
```

macOS (menggunakan Homebrew):

```
brew install node
```

```
node -v
```

```
npm -v
```

2. Setup Proyek Node.js

```
npm init -y
```

```
npm install express mysql bcrypt body-parser cors
```

```
node app.js
```

STRUKTUR DATABASE

Tabel Users

```
CREATE TABLE users (  
    user_id INT PRIMARY KEY AUTO_INCREMENT,  
    full_name VARCHAR(100) NOT NULL,  
    email VARCHAR(100) NOT NULL UNIQUE,  
    phone VARCHAR(20) NOT NULL UNIQUE,  
    password VARCHAR(255) NOT NULL,  
    username VARCHAR(50) UNIQUE,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

Tabel Registrations

```
CREATE TABLE registrations (  
    registration_id INT PRIMARY KEY AUTO_INCREMENT,  
    user_id INT NOT NULL,  
    address TEXT NOT NULL,  
    package_id INT NOT NULL,
```

```
registration_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  
FOREIGN KEY (user_id) REFERENCES users(user_id),  
  
FOREIGN KEY (package_id) REFERENCES subscription_packages(package_id)  
  
);
```

Tabel Subscription Packages

```
CREATE TABLE subscription_packages (  
  
    package_id INT PRIMARY KEY AUTO_INCREMENT,  
  
    package_name VARCHAR(50) NOT NULL,  
  
    price DECIMAL(10, 2) NOT NULL,  
  
    billing_cycle ENUM('monthly', 'yearly') DEFAULT 'monthly',  
  
    speed VARCHAR(20),  
  
    support_type VARCHAR(50),  
  
    mobile_app_available BOOLEAN DEFAULT TRUE,  
  
    max_connections INT,  
  
    features TEXT,  
  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
  
);
```

API ENDPOINT

1. Registrasi Pengguna (POST /api/register)

Request Body:

```
{  
  
    "full_name": "John Doe",  
  
    "email": "john@example.com",  
  
    "phone": "08123456789",
```

```
"address": "Jl. Merdeka No.1",  
"package_id": 1  
}
```

Response:

```
{  
  "message": "Pendaftaran berhasil!",  
  "username": "john",  
  "defaultPassword": "User@1234"  
}
```

2. Mendapatkan Daftar Paket (GET /api/packages)

Response:

```
[  
  {  
    "package_id": 1,  
    "package_name": "Basic Plan",  
    "price": 100000,  
    "features": "Unlimited Data, 24/7 Support"  
  }  
]
```

FRONTEND (PHP & JavaScript)

Form Registrasi

```
fetch('/api/packages')  
  .then(response => response.json())  
  .then(data => {
```

```

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

data.forEach(pkg => {

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

  option.value = pkg.package_id;

  option.textContent = `${pkg.package_name} - Rp${pkg.price}`;

  packageSelect.appendChild(option);

});

});

```

Validasi Form

```

function validateForm() {

  const packageId = document.getElementById('packageSelect').value;

  if (!packageId) {

    alert('Silakan pilih paket sebelum mendaftar.');
```

return false;

```

  }

  return true;

}

```

Proses Registrasi Menggunakan API

```

function submitRegistration() {

  if (!validateForm()) return;

  const formData = {

    full_name: document.getElementById('fullName').value,

    email: document.getElementById('email').value,

    phone: document.getElementById('phone').value,

```

```

address: document.getElementById('address').value,
package_id: document.getElementById('packageSelect').value
};

fetch('/api/register', {
  method: 'POST',
  headers: { 'Content-Type': 'application/json' },
  body: JSON.stringify(formData)
})
.then(response => response.json())
.then(data => {
  if (data.defaultPassword && data.username) {
    alert(`Pendaftaran berhasil! Username: ${data.username}, Password:
    ${data.defaultPassword}`);
  } else {
    alert('Pendaftaran berhasil!');
  }
});
}

```

FITUR UTAMA

- Registrasi dengan validasi otomatis.
- Integrasi API Node.js dan MySQL.
- Penanganan user yang sudah terdaftar.
- Pembuatan username & password default otomatis.

Dokumentasi ini terakhir diperbarui pada: 7 Februari 2025.

