

# **RANCANGAN SISTEM ENTERPRISE RESOURCES PLANNING YAYASAN PENDIDIKAN BERBASIS WEB**



**Oleh :**

Ahmad Hapizhudin (312210370)

## I. Pendahuluan

Rekayasa Perangkat Lunak: Penjelasan, Sejarah, Metode, dan Perkembangannya

## II. Analisa dan Perancangan Sistem

### A. Pendekatan Terstruktur (Struktur)

#### 1. Penjelasan Pendekatan Terstruktur

### B. Pendekatan Berorientasi Objek (OO)

#### 1. Penjelasan Pendekatan Berorientasi Objek (OO)

## III. Unified Modeling Language (UML)

Penjelasan mengenai UML

## IV. Program Desain

Penjelasan Program Desain

## V. Project RPL

# RANCANGAN SISTEM ENTERPRISE RESOURCES PLANNING YAYASAN PENDIDIKAN BERBASIS WEB

## VI. Software Development Life Cycle (SDLC)

### A. Tahap Analisa

#### a. Permasalahan dalam Tahap Analisa

#### b. Kebutuhan Sistem dalam Tahap Analisa

#### c. Use Case Diagram dalam Tahap Analisa

#### d. Activity Diagram dalam Tahap Analisa

#### e. Sequence Diagram dalam Tahap Analisa

### B. Tahap Perancangan atau Desain

#### a. Class Diagram dalam Tahap Perancangan

#### b. Deployment Diagram dalam Tahap Perancangan

#### c. Data Model Diagram dalam Tahap Perancangan

#### d. User Interface dalam Tahap Perancangan

## Pendahuluan

Segala puji dan syukur kita panjatkan ke hadirat Allah, Tuhan Yang Maha Esa, yang dengan rahmat-Nya kita diberikan kesempatan untuk menjalani kehidupan ini. Shalawat serta salam tak terhingga senantiasa kita curahkan kepada Nabi Muhammad SAW, utusan Allah yang membawa cahaya petunjuk bagi umat manusia.

Rekayasa Perangkat Lunak (RPL) adalah sebuah disiplin ilmu dan praktik yang telah mengubah wajah dunia modern yang kita kenal saat ini. Seiring dengan pesatnya perkembangan teknologi, RPL menjadi pondasi bagi penciptaan aplikasi perangkat lunak yang inovatif dan esensial bagi berbagai aspek kehidupan. Dengan tekad untuk merancang, mengembangkan, dan memelihara perangkat lunak yang andal, efisien, serta berdaya guna, RPL membawa kita ke era digital yang revolusioner.

RPL merupakan seni dan ilmu dalam menciptakan solusi perangkat lunak yang mengatasi berbagai permasalahan kompleks di berbagai bidang, seperti bisnis, kesehatan, pendidikan, dan banyak lagi. Dengan landasan prinsip-prinsip terstruktur dan berorientasi objek, RPL memungkinkan kita untuk mewujudkan berbagai aplikasi yang memudahkan kehidupan, meningkatkan produktivitas, dan menyediakan layanan yang lebih baik bagi masyarakat.

Dalam perjalanan ini, kami akan menjelajahi berbagai aspek RPL, termasuk sejarahnya, metodenya, serta perkembangannya yang pesat. Semoga pemahaman mengenai RPL ini dapat mengilhami kita untuk terus berkembang dan berkontribusi dalam dunia perangkat lunak yang terus berubah. Dengan pertolongan Allah dan petunjuk-Nya, mari kita memulai perjalanan ini menuju pemahaman yang lebih mendalam tentang rekayasa perangkat lunak.

# Rekayasa Perangkat Lunak

## 1. Pengertian Perangkat Lunak

Rekayasa perangkat lunak adalah disiplin ilmu yang berkaitan dengan pengembangan, perancangan, dan pemeliharaan perangkat lunak komputer. Ini mencakup seluruh siklus hidup perangkat lunak, mulai dari perumusan kebutuhan, perancangan, implementasi, pengujian, distribusi, pemeliharaan, hingga pensiun perangkat lunak. Berikut ini penjelasan mengenai sejarah, metode, dan perkembangan rekayasa perangkat lunak secara jelas:

## 2. Sejarah Rekayasa Perangkat Lunak:

Sejarah rekayasa perangkat lunak dimulai pada tahun 1940-an ketika komputer pertama diciptakan. Pada awalnya, pengembangan perangkat lunak lebih bersifat ad hoc dan terfokus pada pemrograman tingkat rendah untuk tugas-tugas khusus. Pada tahun 1960-an, industri perangkat lunak semakin berkembang, dan konsep pengembangan perangkat lunak secara sistematis mulai muncul. Pada tahun 1970-an, sejumlah metodologi pengembangan perangkat lunak pertama kali dikembangkan, seperti Waterfall Model, yang memberikan struktur lebih formal kepada proses pengembangan perangkat lunak.

## 3. Metode Rekayasa Perangkat Lunak:

Ada banyak metode yang digunakan dalam rekayasa perangkat lunak. Beberapa di antaranya adalah:

- **Model Air Terjun (Waterfall Model):** Metode ini menggambarkan siklus hidup perangkat lunak sebagai urutan tahapan linear yang meliputi analisis, desain, implementasi, pengujian, dan pemeliharaan.
- **Model Prototipe (Prototyping Model):** Metode ini fokus pada pembuatan prototipe perangkat lunak yang dapat digunakan untuk merancang kebutuhan pengguna.
- **Model Iteratif dan Inkremental (Iterative and Incremental Model):** Pendekatan ini melibatkan pengembangan perangkat lunak dalam serangkaian iterasi, dengan setiap iterasi menambahkan fungsionalitas baru ke produk.
- **Metode Spiral (Spiral Model):** Metode ini menggabungkan aspek iteratif dan inkremental dengan elemen manajemen risiko.
- **Metode Scrum:** Metode ini adalah salah satu pendekatan Agile yang fokus pada kolaborasi tim dan pengiriman perangkat lunak dalam iterasi pendek yang disebut "sprint."

## 4. Perkembangan Rekayasa Perangkat Lunak:

Rekayasa perangkat lunak terus berkembang seiring berjalannya waktu. Beberapa tren dan perkembangan penting termasuk:

**Pendekatan Agile:** Metode Agile, seperti Scrum dan Kanban, semakin populer karena mereka menekankan kolaborasi tim, fleksibilitas, dan pengiriman iteratif.

**DevOps:** DevOps adalah pendekatan yang mengintegrasikan pengembangan perangkat lunak (Dev) dan operasi (Ops) untuk meningkatkan pengiriman perangkat lunak, pemeliharaan, dan kualitas.

**Penggunaan Cloud Computing:** Penggunaan infrastruktur cloud telah memungkinkan pengembang untuk mengakses sumber daya yang fleksibel dan skalabel untuk pengembangan perangkat lunak.

**Kontainerisasi dan Orkestrasi:** Teknologi seperti Docker dan Kubernetes telah memungkinkan pengembang untuk mengelola dan mendistribusikan aplikasi dalam lingkungan yang terisolasi dan terkelola dengan baik.

**Kecerdasan Buatan (AI) dan Machine Learning:** Teknologi ini digunakan dalam pengembangan perangkat lunak untuk meningkatkan analisis data, otomatisasi tugas, dan meningkatkan pengalaman pengguna.

Rekayasa perangkat lunak terus mengalami perubahan dan perkembangan seiring dengan perkembangan teknologi dan kebutuhan pasar. Penting untuk selalu mengikuti tren dan terus memperbarui pengetahuan dalam bidang ini agar dapat menghasilkan perangkat lunak yang efisien, andal, dan relevan dengan kebutuhan pengguna.

## Analisa dan Perancangan Sistem

Analisis dan perancangan sistem dengan pendekatan terstruktur adalah proses yang digunakan dalam pengembangan sistem informasi atau perangkat lunak untuk memastikan bahwa sistem yang dihasilkan memiliki struktur yang terorganisasi dengan baik dan dapat memenuhi kebutuhan bisnis atau pengguna. Pendekatan terstruktur ini menekankan penggunaan teknik dan alat pemodelan yang membantu mengidentifikasi kebutuhan, merancang struktur sistem dengan logis, dan memastikan efisiensi serta kemudahan pemeliharaan. Berikut adalah penjelasan lebih rinci tentang analisis dan perancangan sistem dengan pendekatan terstruktur:

### 1. Analisis Sistem:

Pada tahap analisis sistem, fokus utama adalah memahami kebutuhan bisnis atau organisasi yang akan diakomodasi oleh sistem. Beberapa langkah yang dilakukan dalam analisis sistem meliputi:

**Identifikasi Kebutuhan:** Pengumpulan informasi dari pemangku kepentingan untuk memahami persyaratan sistem. Ini melibatkan wawancara, analisis dokumen, dan diskusi untuk menentukan apa yang seharusnya dicapai oleh sistem.

**Modelisasi Proses Bisnis:** Dokumentasi dan pemodelan proses bisnis yang ada, termasuk alur kerja, peran, aktivitas, dan interaksi antara proses.

**Identifikasi Data:** Mengidentifikasi dan mendokumentasikan data yang diperlukan oleh sistem, termasuk jenis data, entitas, atribut, dan hubungan antar data.

Analisis Kebutuhan Fungsional dan Non-Fungsional: Menentukan kebutuhan fungsional (apa yang sistem harus lakukan) dan kebutuhan non-fungsional (kriteria kinerja, keamanan, dan lain-lain).

**2. Perancangan Sistem:**

Setelah tahap analisis selesai, perancangan sistem dimulai. Ini melibatkan merancang arsitektur sistem dan struktur yang akan memenuhi kebutuhan yang telah diidentifikasi. Beberapa aspek perancangan sistem meliputi:

Arsitektur Sistem: Menentukan bagaimana komponen sistem akan berinteraksi, berkomunikasi, dan disusun secara hierarkis. Ini melibatkan pemilihan teknologi, platform, dan infrastruktur yang sesuai.

Desain Antarmuka Pengguna: Membuat tampilan dan antarmuka pengguna yang mudah digunakan, sesuai dengan kebutuhan dan preferensi pengguna.

Desain Basis Data: Mendefinisikan skema basis data, tabel, dan relasi yang akan digunakan untuk menyimpan dan mengelola data sistem.

Pemodelan Aliran Data: Menggambarkan bagaimana data mengalir melalui sistem, termasuk bagaimana data masukan dan keluaran diolah.

Desain Logika Bisnis: Menentukan logika bisnis dan algoritma yang digunakan untuk mengimplementasikan fungsi sistem.

**UML**

Unified Modeling Language (UML) adalah bahasa pemodelan visual yang digunakan untuk merancang, mendokumentasikan, dan mengkomunikasikan desain perangkat lunak. UML adalah standar industri yang memungkinkan para profesional di bidang rekayasa perangkat lunak untuk menggambarkan konsep, struktur, interaksi, dan perilaku sistem perangkat lunak dengan jelas dan secara terstandarisasi.

UML menggambarkan berbagai aspek perangkat lunak, termasuk diagram untuk menggambarkan struktur (seperti diagram kelas, diagram objek, dan diagram komponen) serta diagram untuk menggambarkan perilaku (seperti diagram aktivitas, diagram urutan, dan diagram kolaborasi). UML juga mencakup konsep pewarisan, asosiasi, agregasi, komposisi, dan banyak elemen lain yang membantu pemodelan perangkat lunak.

Penggunaan UML membantu dalam pengembangan perangkat lunak dengan cara yang lebih terstruktur, serta memungkinkan komunikasi yang lebih baik antara anggota tim pengembangan dan pemangku kepentingan. Ini memainkan peran penting dalam rekayasa perangkat lunak yang efisien dan berhasil.

## Program design

Desain program, yang sering disebut juga sebagai perancangan perangkat lunak (software design), adalah proses mendefinisikan struktur, komponen, dan fitur perangkat lunak yang akan dikembangkan. Ini merupakan tahap penting dalam siklus pengembangan perangkat lunak dan berfokus pada cara perangkat lunak akan diimplementasikan setelah tahap analisis. Berikut adalah beberapa konsep penting dalam desain program:

1. **Arsitektur Perangkat Lunak:** Ini melibatkan pemilihan struktur umum perangkat lunak, seperti arsitektur berorientasi layanan (SOA), arsitektur berbasis mikro, atau arsitektur berlapis. Pemilihan arsitektur ini mempengaruhi bagaimana komponen-komponen perangkat lunak akan berinteraksi dan berkomunikasi satu sama lain.
2. **Desain Antarmuka Pengguna (UI/UX):** Ini adalah tentang merancang antarmuka pengguna yang dapat digunakan dengan mudah dan memenuhi kebutuhan pengguna. Ini mencakup perencanaan tampilan, navigasi, interaksi pengguna, dan elemen desain visual.
3. **Desain Basis Data:** Ini melibatkan perancangan struktur basis data yang akan digunakan untuk menyimpan dan mengelola data aplikasi. Ini mencakup pemodelan tabel, relasi, dan pemilihan database yang sesuai.
4. **Desain Algoritma:** Ini adalah tentang merancang algoritma dan logika bisnis yang akan digunakan dalam perangkat lunak. Ini mencakup pemilihan algoritma yang efisien dan efektif untuk menyelesaikan tugas yang diinginkan.
5. **Desain Modul/Component:** Perangkat lunak sering dibagi menjadi berbagai modul atau komponen yang memiliki tugas dan tanggung jawab tertentu. Desain ini melibatkan pemilihan komponen, pemodelan hubungan antara mereka, dan spesifikasi fungsi masing-masing.
6. **Perancangan Pengujian:** Ini mencakup merancang rencana pengujian untuk memastikan bahwa perangkat lunak berfungsi dengan baik. Ini mencakup pengujian unit, integrasi, fungsional, dan pengujian performa sesuai kebutuhan.
7. **Pemilihan Teknologi:** Pemilihan teknologi dan alat yang sesuai untuk pengembangan perangkat lunak. Ini mencakup bahasa pemrograman, framework, perangkat keras, dan perangkat lunak pendukung lainnya.
8. **Desain Keamanan:** Ini melibatkan pemikiran tentang keamanan perangkat lunak, termasuk pencegahan serangan siber dan perlindungan data sensitif.
9. **Pemikiran tentang Pemeliharaan:** Desain program juga harus mempertimbangkan pemeliharaan jangka panjang, termasuk cara mengelola perubahan, peningkatan, dan perbaikan.
10. **Desain program adalah langkah kunci dalam pengembangan perangkat lunak yang berperan dalam mengubah konsep dari tahap analisis menjadi perangkat lunak yang nyata. Itu membantu memastikan**

bahwa perangkat lunak dikembangkan dengan efisien, mudah dipelihara, dan sesuai dengan kebutuhan pengguna.

## **RANCANGAN SISTEM ENTERPRISE RESOURCES PLANNING YAYASAN PENDIDIKAN BERBASIS WEB**

Dalam era digital yang semakin maju, pengelolaan Yayasan Pendidikan memerlukan system yang efisien, terintegrasi, dan dapat diakses secara online untuk meningkatkan efektivitas operasional dan pengalaman pengguna. Enterprise Resources Planning (ERP) adalah solusi yang memungkinkan Yayasan Pendidikan untuk mengintegrasikan, mengelola, dan mengoptimalkan berbagai aspek operasional mereka, dan manajemen akademik hingga keuangan dan sumber data manusia.

### **a. Permasalahan dalam system ini**

Dalam merancang dan mengimplementasikan Sistem ERP Yayasan Pendidikan Berbasis Web, ada beberapa permasalahan dan tantangan yang mungkin dihadapi. Berikut adalah beberapa permasalahan yang dapat muncul selama proses pengembangan dan pengoperasian sistem tersebut:

#### **1. Ketidaksesuaian dengan Kebutuhan Pengguna:**

Salah satu permasalahan utama adalah ketidaksesuaian sistem dengan kebutuhan sebenarnya pengguna, seperti staf administrasi, guru, siswa, dan orang tua. Sistem harus dirancang sedemikian rupa agar sesuai dengan kebutuhan dan preferensi mereka.

#### **2. Integrasi Data yang Rumit:**

Yayasan pendidikan mungkin telah mengumpulkan data dari berbagai sumber yang berbeda. Integrasi data yang rumit dan penyelarasan data yang akurat dari berbagai sumber menjadi tantangan yang signifikan.

#### **3. Keamanan Data:**

Keamanan data siswa, staf, dan informasi keuangan merupakan masalah kritis. Peretasan data atau kebocoran informasi pribadi dapat mengakibatkan dampak serius.

#### **4. Skalabilitas:**

Sistem harus mampu menangani pertumbuhan jumlah pengguna dan data seiring berjalannya waktu. Permasalahan skalabilitas perlu diatasi untuk memastikan sistem tetap berjalan lancar.

#### **5. Ketersediaan Sistem:**

Ketersediaan sistem yang rendah dapat mengganggu operasional yayasan pendidikan. Permasalahan teknis atau pemeliharaan yang tidak terjadwal dapat mengakibatkan gangguan layanan.

#### **6. Pelatihan Pengguna:**

Membekali pengguna (staf, guru, siswa, dan orang tua) dengan pemahaman yang memadai tentang cara menggunakan sistem adalah permasalahan penting. Pelatihan yang efektif perlu diselenggarakan.

#### **7. Pemeliharaan dan Dukungan:**



Sistem perlu dipelihara secara berkala dan dukungan teknis harus tersedia. Permasalahan yang muncul harus ditangani dengan cepat dan efisien.

8. Kepatuhan Regulasi:

Yayasan pendidikan perlu mematuhi regulasi dan kebijakan yang berlaku dalam pengelolaan data siswa dan operasional mereka. Melanggar regulasi dapat berakibat hukuman dan sanksi yang serius.  
Biaya Pengembangan dan Operasional:

Mengembangkan dan menjaga sistem ERP dapat mahal. Permasalahan terkait anggaran harus dikelola dengan bijak.

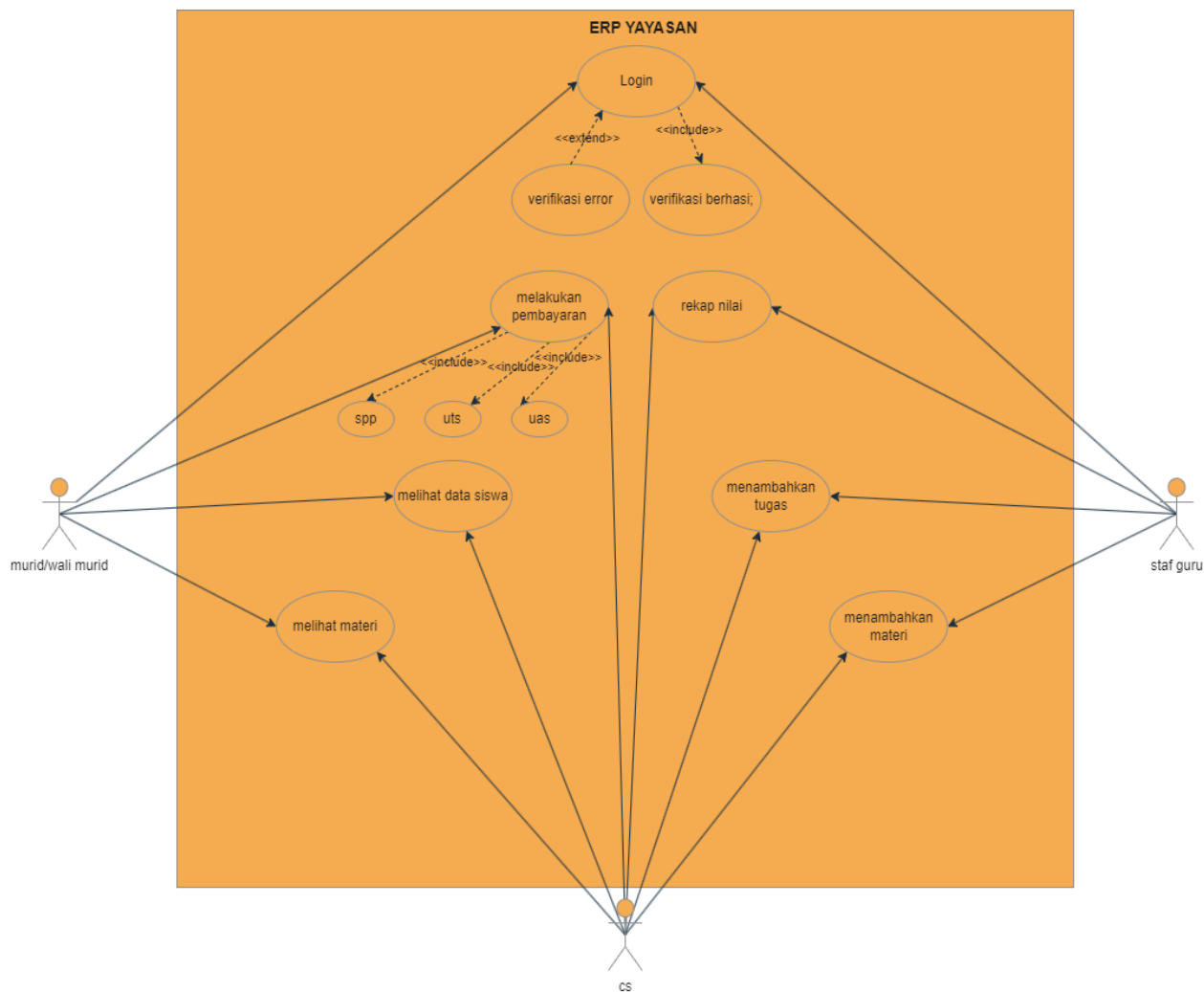
9. Perubahan Kebijakan dan Kebutuhan Bisnis:

Yayasan pendidikan mungkin mengalami perubahan kebijakan dan kebutuhan bisnis seiring berjalannya waktu. Sistem harus dapat menyesuaikan diri dengan perubahan ini.

Permasalahan ini memerlukan perencanaan, perhatian, dan solusi yang matang selama pengembangan dan pengoperasian sistem ERP Yayasan Pendidikan Berbasis Web. Dengan perencanaan yang baik dan pemahaman yang mendalam tentang kebutuhan pengguna, banyak dari permasalahan ini dapat diatasi dengan efektif.

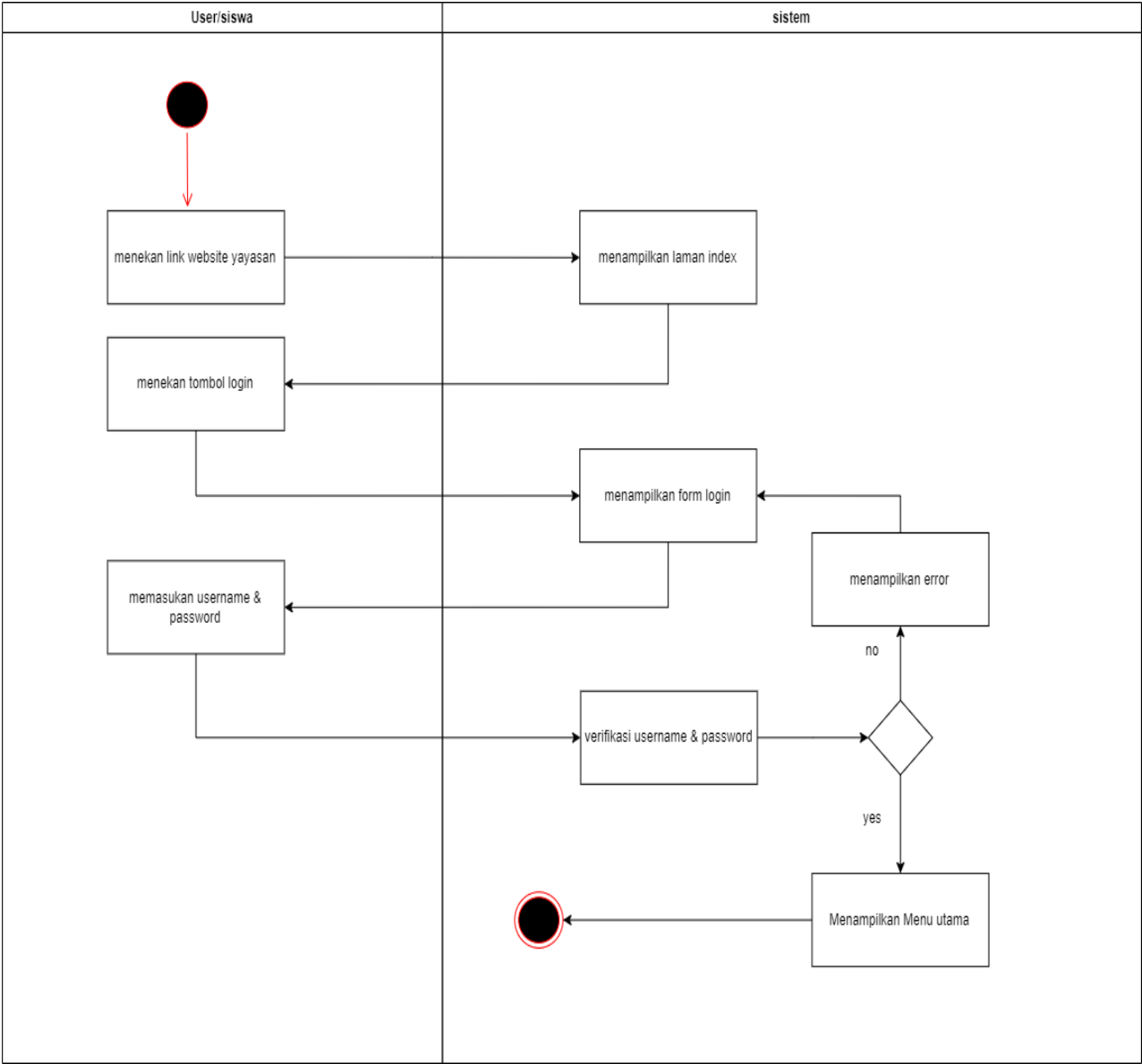
b. Kebutuhan System

c. Use Case Diagram

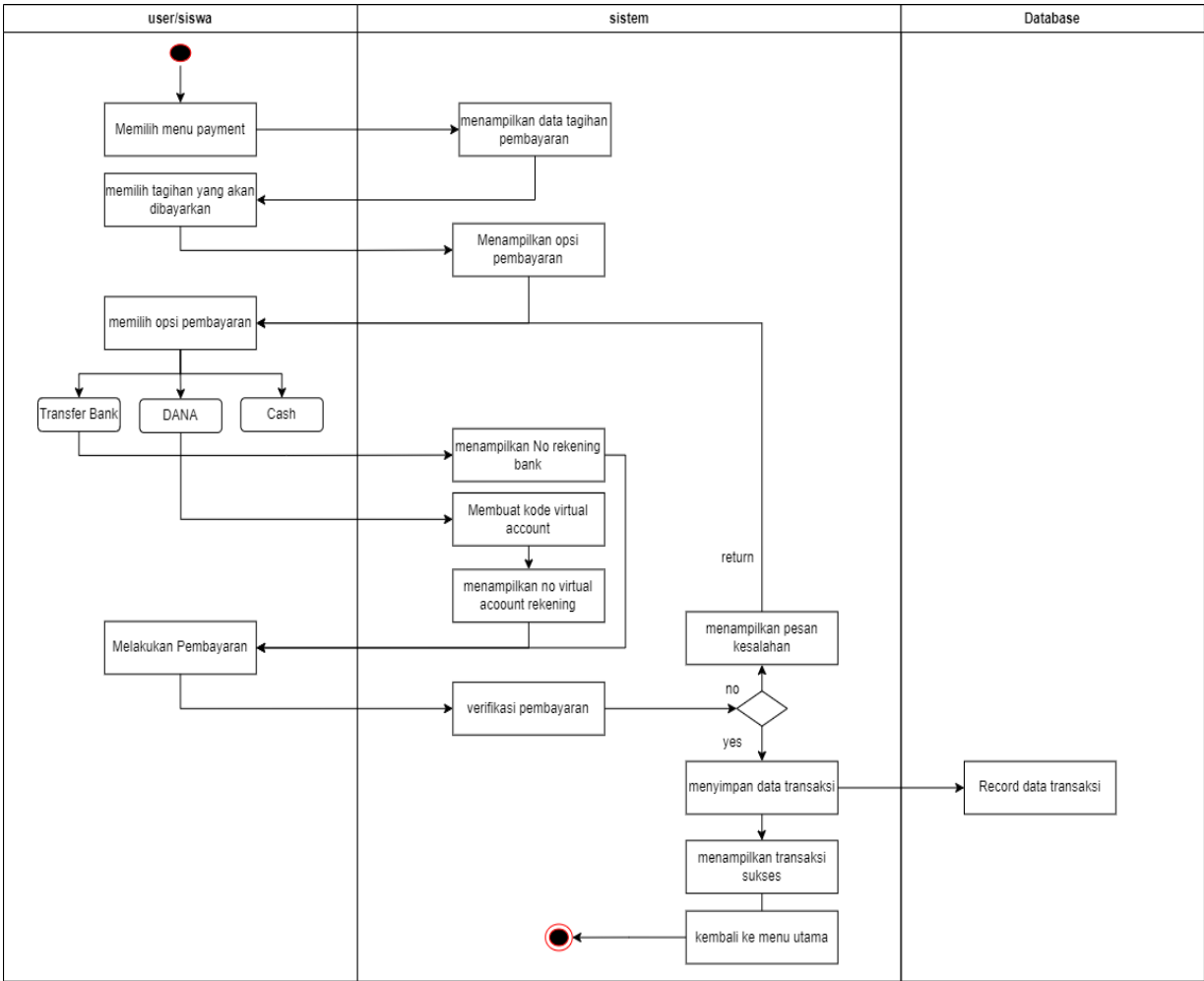


d. Activity Diagram

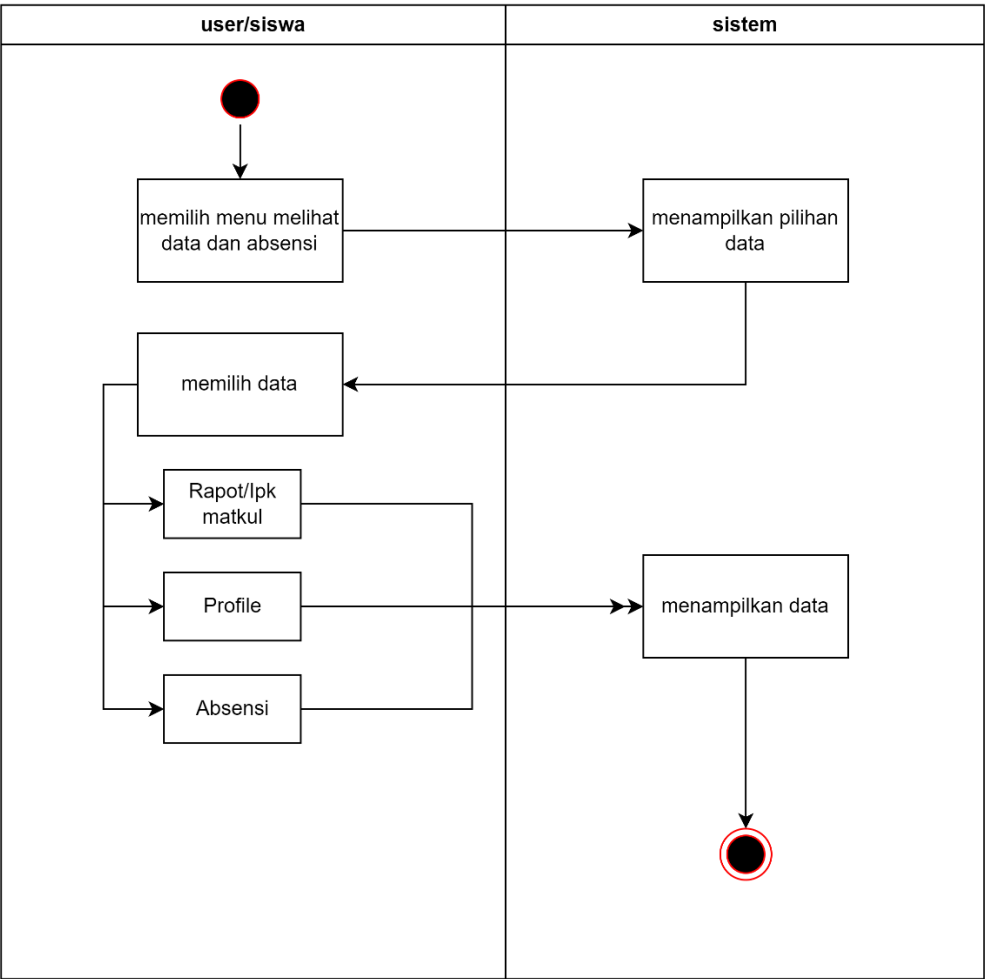
Activity Login



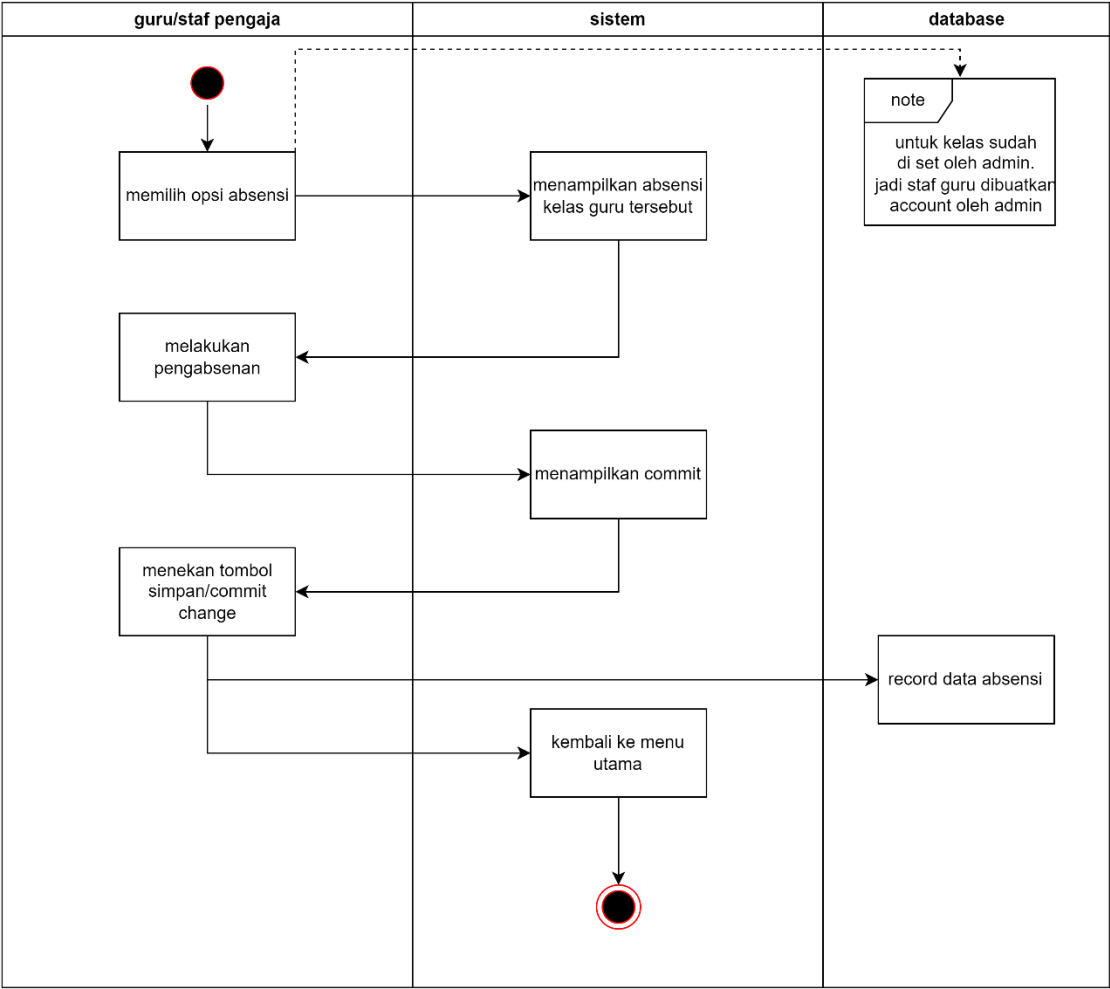
Activity Payment/ Pembayaran



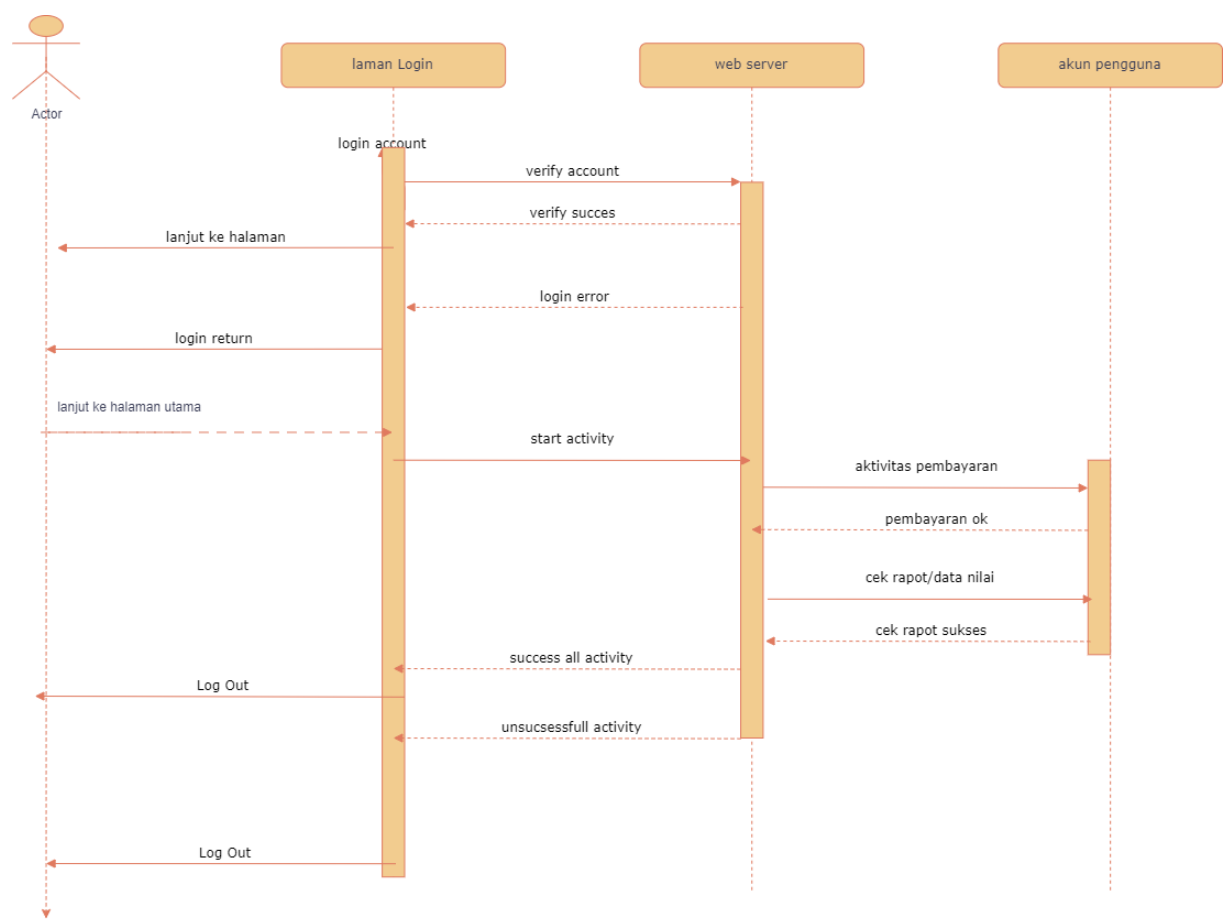
**Activity Acces raport dan nilai**



Activity pengabsenan

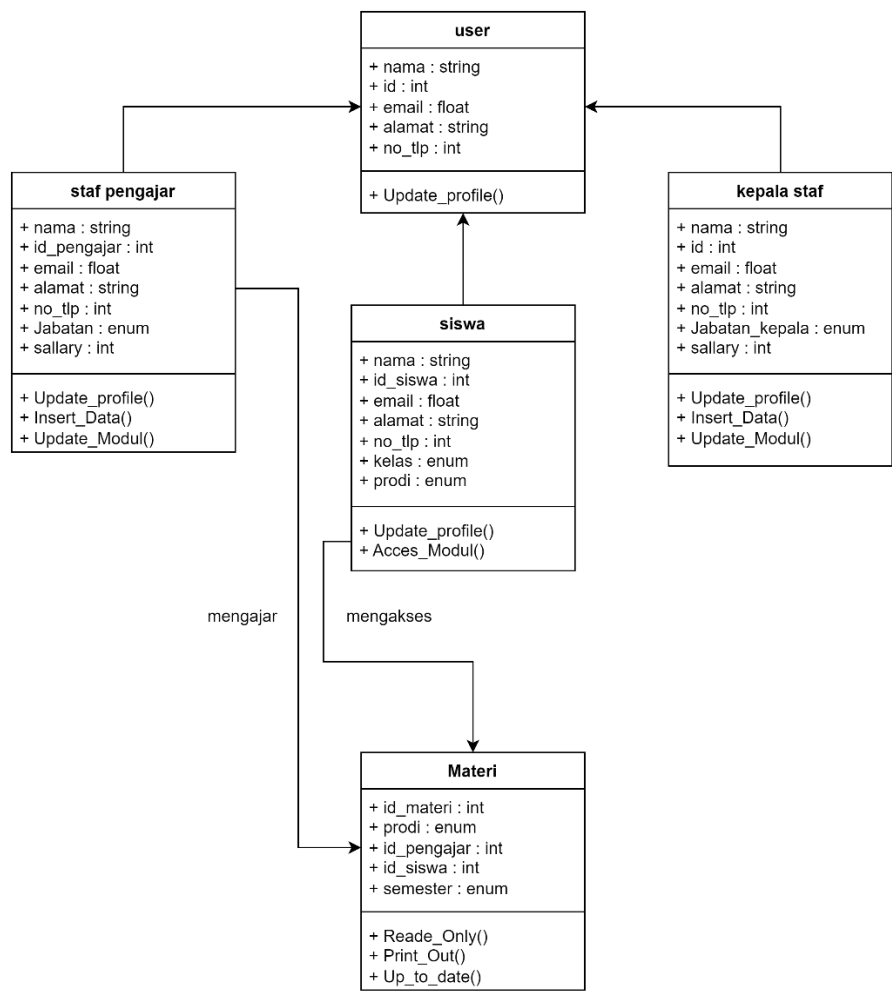


e. Sequence Diagram

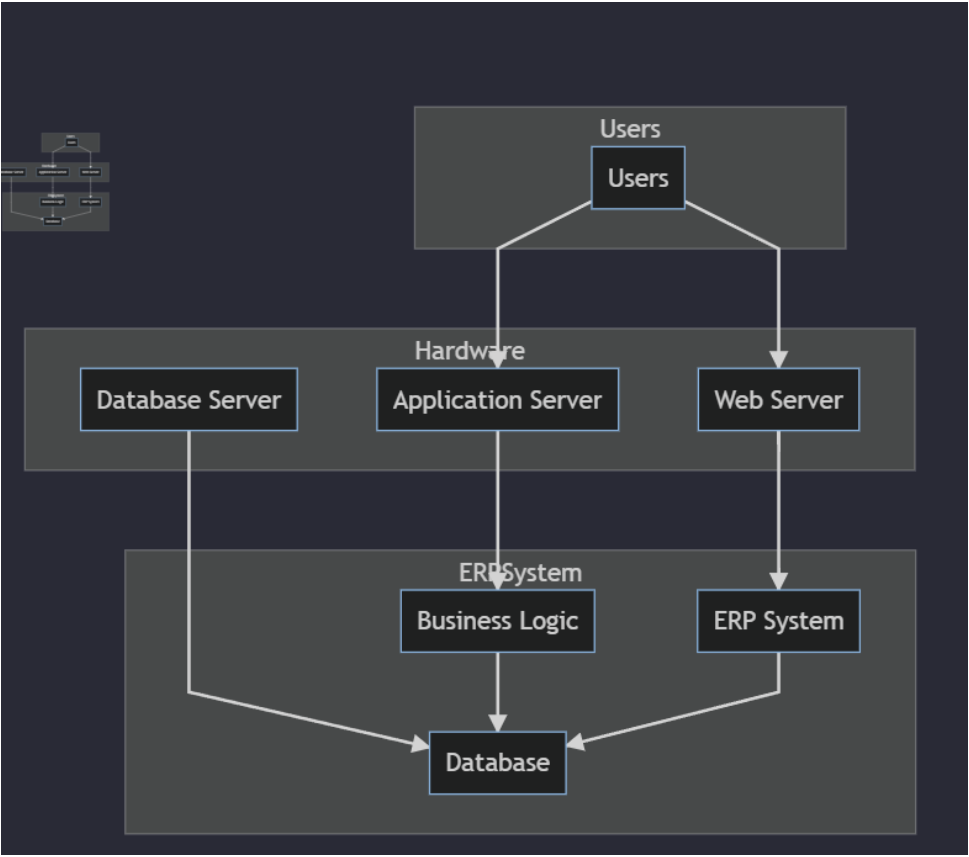


Perancangan dan Design

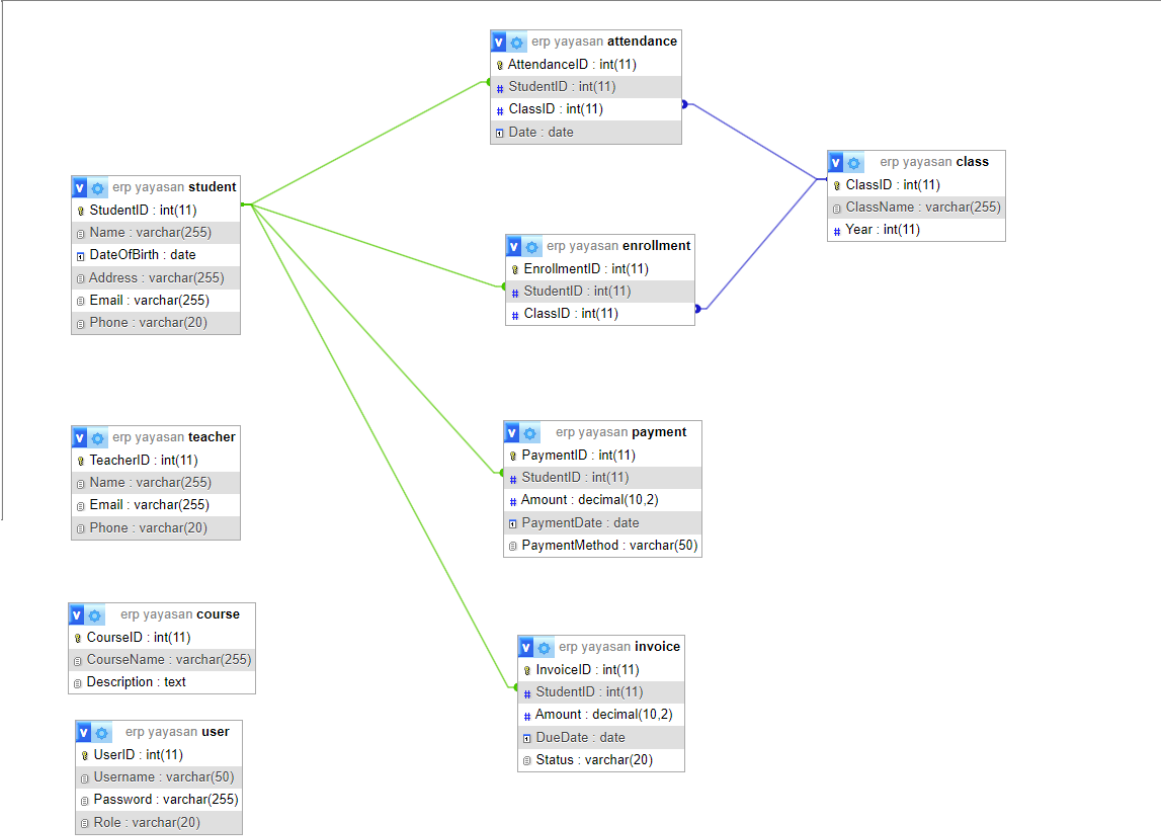
a. Class Diagram



b. Deployment Diagram



c. Data Model Diagram



d. User Interface