

# **REKAYASA PERANGKAT LUNAK**



**Rico Prasetya**

**312210425**

**TI.22.A.1**

**Dosen Pengampu :**

**Wahyu Hadikristanto, S.Kom, M.Kom**

**Teknik Informatika**

**Universitas Pelita Bangsa**

**2023**

Nama Project : Sistem Informasi Kelas

1. Berikan penjelasan mengenai Rekayasa Perangkat lunak, sejarahnya, metodenya dan perkembangannya!

Jawaban :

Rekayasa Perangkat Lunak (Software Engineering) adalah disiplin ilmu yang berkaitan dengan pengembangan, perancangan, pemeliharaan, dan manajemen perangkat lunak atau software. Ini mencakup berbagai aspek yang berkaitan dengan perangkat lunak, termasuk analisis kebutuhan, perancangan, pengkodean, pengujian, dokumentasi, dan pemeliharaan.

Berikut penjelasan lebih rinci tentang Rekayasa Perangkat Lunak, termasuk sejarahnya, metodenya, dan perkembangannya:

- a. Sejarah Rekayasa Perangkat Lunak:  
Sejarah rekayasa perangkat lunak dimulai pada tahun 1960-an ketika perangkat lunak mulai menjadi komponen kritis dalam sistem komputer. Pada awalnya, pengembangan perangkat lunak dilakukan secara tidak terstruktur, yang mengakibatkan kesulitan dalam pemeliharaan dan manajemen perangkat lunak. Pada tahun 1968, pertemuan di Garmisch, Jerman, menghasilkan konferensi pertama yang secara resmi menyebutkan "rekayasa perangkat lunak."
- b. Metode dalam Rekayasa Perangkat Lunak:  
Ada banyak metode yang digunakan dalam rekayasa perangkat lunak. Beberapa metode yang umum digunakan termasuk:
  - i. Metode Waterfall: Ini adalah pendekatan linear yang mengatur proses pengembangan perangkat lunak menjadi serangkaian tahap seperti analisis, perancangan, pengkodean, pengujian, dan pemeliharaan.
  - ii. Metode Agile: Ini adalah pendekatan yang lebih fleksibel dan berfokus pada kerja tim kolaboratif, iterasi, dan respons cepat terhadap perubahan kebutuhan pengguna.
  - iii. Metode Scrum: Ini adalah kerangka kerja manajemen proyek Agile yang berfokus pada pengaturan waktu dalam periode waktu tetap yang disebut sprint.
  - iv. Metode DevOps: Ini adalah pendekatan yang menggabungkan pengembangan dan operasi untuk mempercepat pengiriman perangkat lunak dan meningkatkan kerja sama antara tim pengembangan dan tim operasi.
- c. Perkembangan Rekayasa Perangkat Lunak:  
Rekayasa perangkat lunak terus berkembang seiring waktu. Perkembangan utama termasuk:
  - i. Peningkatan alat dan teknologi: Perkembangan dalam alat pengembangan perangkat lunak, bahasa pemrograman, dan kerangka kerja telah memungkinkan pengembang untuk lebih efisien dan produktif.
  - ii. Cloud computing: Pengembangan teknologi cloud computing telah memungkinkan penyediaan perangkat lunak sebagai layanan (SaaS) dan pengelolaan infrastruktur yang lebih efisien.

- iii. Penerapan kecerdasan buatan (AI) dalam pengembangan perangkat lunak: AI digunakan untuk otomatisasi tugas-tugas pengembangan, pengujian, dan pemeliharaan perangkat lunak.
- iv. Peningkatan fokus pada keamanan perangkat lunak: Dengan ancaman keamanan yang semakin kompleks, keamanan perangkat lunak telah menjadi perhatian utama dalam rekayasa perangkat lunak.

Rekayasa perangkat lunak terus berkembang dan beradaptasi dengan perubahan teknologi dan kebutuhan bisnis. Ini tetap menjadi bagian integral dalam perkembangan teknologi informasi dan memiliki peran penting dalam keberhasilan produk dan layanan perangkat lunak di berbagai industri.

## 2. Jelaskan mengenai Analisa dan Perancangan sistem dengan pendekatan terstruktur(Struktur).

Jawaban :

Analisis dan perancangan sistem dengan pendekatan terstruktur, atau sering disebut sebagai pendekatan berstruktur (Structured System Analysis and Design) adalah metode yang digunakan dalam rekayasa perangkat lunak untuk mengidentifikasi, merencanakan, dan merancang sistem informasi dengan cara yang terorganisir dan terstruktur. Pendekatan ini berfokus pada pemahaman yang mendalam tentang masalah yang akan diselesaikan oleh sistem, serta pembuatan desain yang sistematis dan terstruktur untuk memenuhi kebutuhan tersebut. Berikut adalah penjelasan lebih lanjut tentang analisis dan perancangan sistem dengan pendekatan terstruktur:

- a. Analisis Sistem: Tahap analisis sistem adalah langkah awal dalam pengembangan sistem. Tujuannya adalah untuk memahami masalah atau kebutuhan yang akan diatasi oleh sistem yang akan dikembangkan. Beberapa langkah dalam analisis sistem meliputi:
  - i. Identifikasi stakeholder (pemangku kepentingan) dan kebutuhan mereka.
  - ii. Pengumpulan data dan informasi terkait dengan masalah yang ada.
  - iii. Pembuatan model yang menggambarkan alur informasi, proses bisnis, dan hubungan antara entitas yang ada.
  - iv. Identifikasi masalah dan peluang perbaikan.
- b. Perancangan Sistem: Setelah analisis sistem selesai, tahap perancangan sistem dimulai. Tujuannya adalah untuk merancang solusi sistem yang akan memenuhi kebutuhan yang telah diidentifikasi. Beberapa langkah dalam perancangan sistem meliputi:
  - i. Menentukan struktur umum sistem dan mengidentifikasi komponen utama.
  - ii. Membangun model desain yang mencakup tampilan antarmuka pengguna, struktur data, dan algoritma pemrosesan.
  - iii. Mengidentifikasi perangkat keras dan perangkat lunak yang dibutuhkan untuk mengimplementasikan sistem.
  - iv. Memastikan bahwa desain sistem memenuhi kebutuhan pengguna dan tujuan proyek.
- c. Pendekatan Terstruktur: Pendekatan terstruktur menggunakan notasi dan teknik yang terorganisir untuk mewakili analisis dan desain sistem. Salah satu notasi yang sering digunakan adalah Diagram Alur Data (Data Flow Diagram - DFD) yang menggambarkan aliran data dalam sistem. DFD digunakan untuk mengidentifikasi

entitas, proses, aliran data, dan penyimpanan data dalam sistem. Selain DFD, pendekatan terstruktur juga mencakup Diagram Struktur (Structure Chart) yang digunakan untuk menggambarkan struktur hierarki dalam sistem dan bagaimana komponen-komponen berinteraksi.

d. Keuntungan Pendekatan Terstruktur:

- i. Memudahkan pemahaman dan dokumentasi sistem.
- ii. Memungkinkan identifikasi masalah dan peluang perbaikan dengan lebih jelas.
- iii. Menghasilkan desain sistem yang lebih terstruktur dan terorganisir.
- iv. Memudahkan pemeliharaan dan perubahan sistem di masa depan.

Pendekatan terstruktur merupakan pendekatan yang cukup klasik dalam rekayasa perangkat lunak dan telah digunakan secara luas selama beberapa dekade. Meskipun sejumlah metodologi pengembangan perangkat lunak modern seperti metode Agile telah muncul, pendekatan terstruktur masih memiliki relevansi dalam pengembangan sistem yang kompleks dan besar.

3. Jelaskan mengenai Analisa dan Perancangan sistem dengan pendekatan berorientasi object(OO)!

Jawaban :

Analisis dan Perancangan Sistem dengan pendekatan berorientasi objek (OO), atau sering disebut Analisis dan Perancangan Berorientasi Objek (Object-Oriented Analysis and Design, OOAD), adalah metode pengembangan perangkat lunak yang berfokus pada penggunaan konsep objek untuk menganalisis, merancang, dan mengembangkan sistem. Pendekatan ini berdasarkan konsep objek yang merupakan abstraksi dari entitas yang memiliki atribut (data) dan metode (fungsi atau operasi) yang dapat dijalankan pada objek tersebut. Berikut adalah penjelasan lebih rinci tentang analisis dan perancangan sistem dengan pendekatan berorientasi objek:

a. Analisis Sistem Berorientasi Objek:

- i. Identifikasi Objek: Dalam tahap analisis sistem berorientasi objek, fokus utama adalah mengidentifikasi objek-objek yang relevan dalam domain masalah atau kebutuhan bisnis. Objek-objek ini mungkin mencakup entitas, konsep, atau komponen yang ada dalam sistem yang akan dikembangkan.
- ii. Model Konsep dan Hubungan: Selama analisis, objek-objek dianalisis lebih lanjut untuk memahami atribut, metode, dan hubungan antar objek. Diagram kelas dan diagram objek sering digunakan untuk merepresentasikan konsep dan hubungan di antara objek-objek.
- iii. Identifikasi Kebutuhan Fungsional dan Non-Fungsional: Selain identifikasi objek, analisis berorientasi objek juga mencakup identifikasi kebutuhan fungsional (apa yang sistem harus lakukan) dan kebutuhan non-fungsional (bagaimana sistem harus melakukannya, seperti performa, keamanan, dan skalabilitas).

b. Perancangan Sistem Berorientasi Objek:

- i. **Pemodelan Desain:** Dalam tahap perancangan, objek-objek ditempatkan dalam sebuah arsitektur sistem yang direncanakan. Ini melibatkan pemodelan struktur kelas dan hubungan antara kelas. Diagram kelas dan diagram relasi objek digunakan untuk menggambarkan desain ini.
    - ii. **Pemetaan Objek ke Kode Pemrograman:** Setelah desain sistem berorientasi objek selesai, langkah selanjutnya adalah mengimplementasikannya dalam bahasa pemrograman. Objek-objek yang telah didefinisikan dalam desain diubah menjadi kode sumber.
  - c. **Konsep Utama Berorientasi Objek:**
    - i. **Kelas dan Objek:** Kelas adalah blueprint untuk menciptakan objek. Objek adalah instansi dari kelas dan menggabungkan data (atribut) dan perilaku (metode).
    - ii. **Enkapsulasi:** Enkapsulasi adalah konsep di mana objek menyembunyikan detail implementasi internalnya dan hanya menyediakan antarmuka untuk berinteraksi dengan dunia luar.
    - iii. **Pewarisan (Inheritance):** Pewarisan memungkinkan kelas baru (subkelas) untuk mewarisi atribut dan metode dari kelas yang ada (superkelas).
    - iv. **Polimorfisme:** Polimorfisme memungkinkan objek dengan jenis yang berbeda untuk merespons metode dengan cara yang sesuai dengan jenis objek tersebut.
  - d. **Keuntungan Pendekatan Berorientasi Objek:**
    - i. Memudahkan pemahaman dan pemeliharaan kode.
    - ii. Memungkinkan penggunaan kembali kode (reusability) melalui pewarisan dan komposisi objek.
    - iii. Memungkinkan model sistem yang lebih sesuai dengan dunia nyata.
    - iv. Mendukung pengembangan sistem yang modular dan skalabel.
- Pendekatan berorientasi objek telah menjadi salah satu paradigma utama dalam pengembangan perangkat lunak, dan banyak bahasa pemrograman seperti Java, C++, dan Python didasarkan pada konsep objek. Pendekatan ini sangat relevan dalam pengembangan aplikasi yang kompleks dan membutuhkan pemodelan yang baik dari entitas dalam domain masalah.

#### 4. Berikan penjelasan mengenai UML!

Jawaban :

Unified Modeling Language (UML) adalah bahasa visual yang digunakan untuk memodelkan, merancang, dan mendokumentasikan sistem perangkat lunak dan proses bisnis. UML adalah standar industri yang dikembangkan oleh Object Management Group (OMG) dan telah menjadi alat yang sangat penting dalam rekayasa perangkat lunak. UML menyediakan sejumlah diagram dan notasi untuk menggambarkan berbagai aspek dari perangkat lunak dan sistem. Berikut adalah penjelasan mengenai UML:

- a. **Tujuan UML:**
  - i. **Modeling:** UML digunakan untuk membuat model yang menggambarkan berbagai aspek sistem perangkat lunak, termasuk struktur, perilaku, interaksi, dan komunikasi antara komponen.

- ii. Communication: UML membantu dalam komunikasi antara tim pengembangan, pemangku kepentingan, dan pemrogram melalui gambaran visual yang jelas.
  - iii. Documentation: UML juga berfungsi sebagai alat dokumentasi yang efektif untuk sistem perangkat lunak, memungkinkan tim pengembangan dan pemeliharaan untuk memahami dan memodifikasi sistem dengan lebih baik.
  - iv. Analysis and Design: UML dapat digunakan dalam analisis dan perancangan sistem berorientasi objek.
- b. Notasi UML:  
UML menggunakan sejumlah notasi, termasuk:
  - i. Diagram Kelas (Class Diagram): Menggambarkan struktur kelas dan hubungan antara kelas dalam sistem.
  - ii. Diagram Use Case (Use Case Diagram): Menggambarkan interaksi antara sistem dan pemangku kepentingan (aktor) serta skenario penggunaan sistem.
  - iii. Diagram Aktivitas (Activity Diagram): Menggambarkan aliran kerja atau aktivitas dalam sistem.
  - iv. Diagram Sekuensial (Sequence Diagram): Menunjukkan bagaimana objek berinteraksi sepanjang waktu dan urutan pesan yang dikirim antara objek.
  - v. Diagram Komponen (Component Diagram): Menggambarkan komponen perangkat lunak dan hubungan antara mereka.
  - vi. Diagram Interaksi (Interaction Diagrams): Ini termasuk Diagram Sekuensial dan Diagram Kolaborasi untuk menggambarkan interaksi antara objek.
  - vii. Diagram Status (Statechart Diagram): Menggambarkan bagaimana objek berperilaku selama berbagai keadaan atau status.
  - viii. Diagram Kegiatan (Activity Diagram): Menggambarkan aliran aktivitas dalam suatu proses.
  - ix. Diagram Waktu (Timing Diagram): Digunakan untuk menggambarkan perubahan keadaan objek dalam interval waktu tertentu.
- c. Komponen UML:  
UML memiliki beberapa komponen dasar, seperti kelas, objek, antarmuka, hubungan, kolaborasi, paket, dan banyak lagi. Setiap komponen memiliki peran dan notasi yang berbeda dalam diagram UML.
- d. Fase Pengembangan: UML dapat digunakan dalam berbagai tahap pengembangan perangkat lunak, termasuk analisis, perancangan, implementasi, dan pemeliharaan.
- e. Manfaat UML:
  - i. Meningkatkan pemahaman sistem: UML membantu dalam memvisualisasikan dan memahami struktur dan perilaku sistem secara lebih baik.
  - ii. Peningkatan komunikasi: UML memfasilitasi komunikasi antara pemangku kepentingan dan anggota tim pengembangan.
  - iii. Dokumentasi yang kuat: UML memungkinkan penciptaan dokumentasi yang kuat untuk sistem perangkat lunak.
  - iv. Reusabilitas dan skalabilitas: UML memfasilitasi pemodelan yang mendukung reusabilitas kode dan skalabilitas sistem.

UML adalah alat yang sangat penting dalam rekayasa perangkat lunak modern dan digunakan secara luas dalam pengembangan sistem perangkat lunak yang kompleks. Ini membantu tim pengembangan dalam memahami, merancang, dan

mengimplementasikan sistem dengan lebih baik, serta memfasilitasi komunikasi yang efektif antara semua pemangku kepentingan proyek.

5. Berikan penjelasan mengenai Program Desain!

Jawaban :

Program desain (design pattern) merujuk pada solusi umum yang telah teruji dan terbukti dalam rekayasa perangkat lunak untuk mengatasi masalah yang sering muncul dalam pengembangan perangkat lunak. Desain pattern adalah konsep yang digunakan untuk memecahkan masalah dalam perangkat lunak dengan cara yang terstruktur, terdokumentasi, dan mudah dimengerti. Mereka membantu dalam menghasilkan kode yang lebih terstruktur, mudah dipahami, dan mudah dipelihara.

Berikut adalah beberapa poin penting dalam menjelaskan program desain:

- a. Tujuan Program Desain:
  - i. Meningkatkan Kualitas Perangkat Lunak: Program desain membantu dalam menciptakan perangkat lunak yang lebih baik dari segi struktur, kualitas, dan pemeliharaan.
  - ii. Reusabilitas: Desain pattern memungkinkan penggunaan kembali solusi yang terbukti untuk masalah yang serupa dalam berbagai proyek.
  - iii. Komunikasi: Mereka berfungsi sebagai bahasa umum yang dapat digunakan untuk berkomunikasi antara anggota tim pengembangan.
  - iv. Keandalan: Menggunakan desain pattern yang terbukti dapat meningkatkan keandalan perangkat lunak.
- b. Contoh Program Desain Terkenal:
  - i. Singleton Pattern: Digunakan untuk memastikan bahwa hanya ada satu instansi dari sebuah kelas dalam sistem.
  - ii. Factory Method Pattern: Membungkus pembuatan objek dalam sebuah metode yang dapat dioverride oleh subkelas.
  - iii. Observer Pattern: Menggunakan hubungan satu-ke-banyak antara objek, sehingga jika objek satu berubah, objek lain yang memerlukannya akan diberi tahu.
  - iv. Strategy Pattern: Memungkinkan penggantian algoritma atau strategi dalam runtime.
  - v. Decorator Pattern: Memungkinkan penambahan perilaku tambahan pada objek tanpa mengubah struktur dasarnya.
  - vi. MVC (Model-View-Controller) Pattern: Membagi aplikasi menjadi tiga komponen terpisah untuk mengatur tampilan, logika bisnis, dan pemodelan data.
- c. Cara Menggunakan Program Desain:
  - i. Program desain umumnya diterapkan dalam bahasa pemrograman tertentu dengan mengikuti prinsip-prinsip yang didefinisikan oleh desain pattern tersebut.
  - ii. Biasanya, desain pattern adalah abstraksi yang tidak memiliki implementasi konkret. Oleh karena itu, pengembang harus

mengimplementasikan desain pattern sesuai dengan kebutuhan dan bahasa pemrograman yang digunakan.

- iii. Buku seperti "Design Patterns: Elements of Reusable Object-Oriented Software" oleh Erich Gamma, Richard Helm, Ralph Johnson, dan John Vlissides (sering disebut sebagai "Gang of Four" atau GoF) adalah sumber yang sangat baik untuk memahami dan mempelajari desain pattern.

d. Evolusi dan Perkembangan Program Desain:

- i. Program desain terus berkembang seiring waktu dengan munculnya desain pattern baru dan perubahan dalam paradigma pemrograman.
- ii. Perpustakaan desain pattern yang terkenal seperti "Pattern-Oriented Software Architecture" (POSA) dan "Enterprise Integration Patterns" memiliki kontribusi besar dalam perkembangan desain pattern.

Program desain adalah alat penting dalam pengembangan perangkat lunak yang membantu dalam menciptakan solusi yang efisien, mudah dimengerti, dan mudah dipelihara. Memahami dan menerapkan desain pattern yang sesuai dapat memperbaiki kualitas dan produktivitas pengembangan perangkat lunak.

6. Tuliskan nama Project RPL Anda!

Jawaban : Sistem Informasi Kelas

7. Dalam SDLC setelah Planning ada tahap Analisa:

- a. Tuliskan Permasalahannya!

Jawaban :

Sistem ini dibuat untuk membantu pendataan agar lebih efisien dan akurat karena system pendataan manual masih memiliki kekurangan yaitu keakuratan data serta rawannya terjadi kehilangan data

- b. Tuliskan Kebutuhan sistemnya!

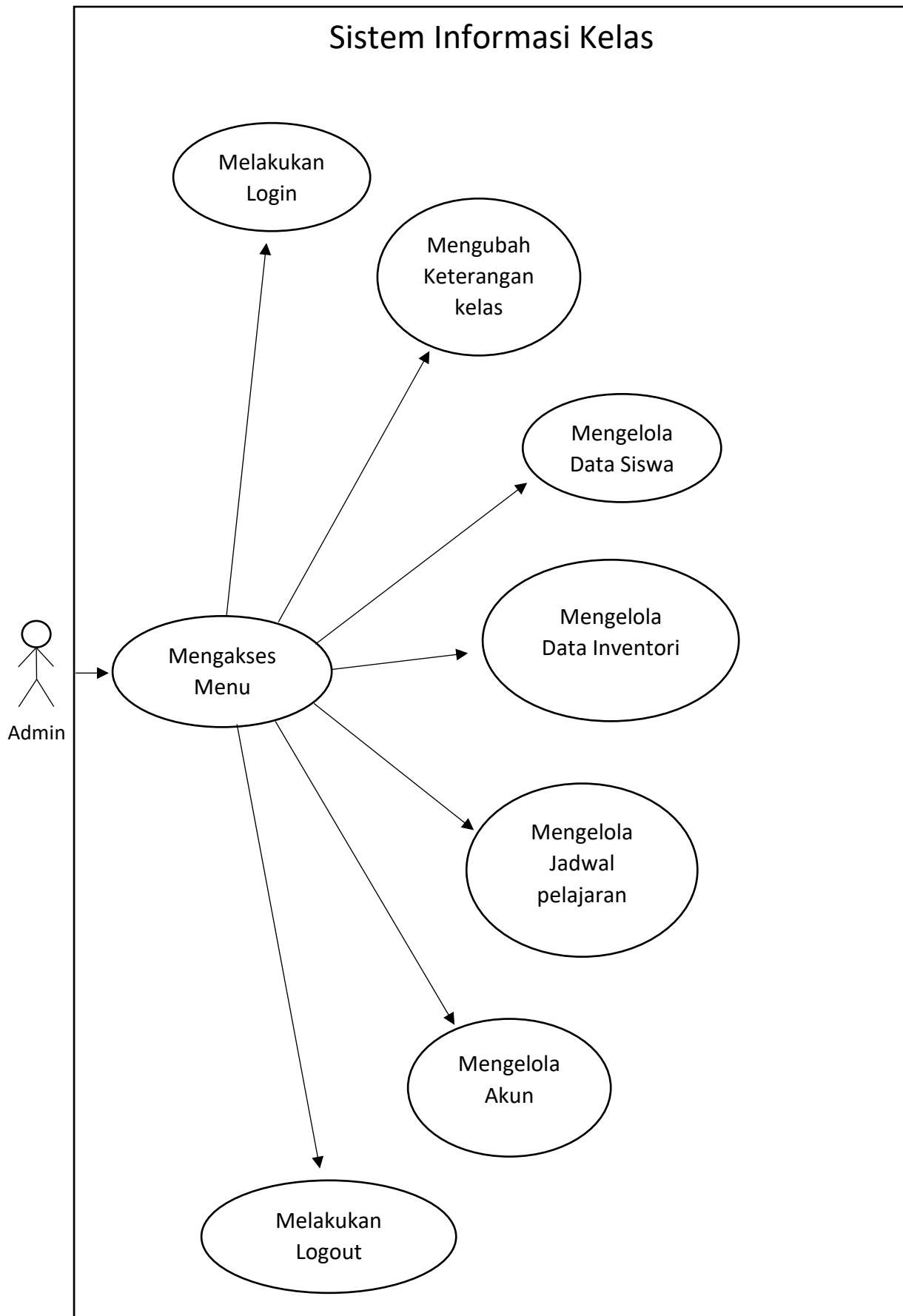
Jawaban :

Kebutuhan Fungsional  
Kebutuhan Antarmuka Pengguna  
Kebutuhan Keamanan  
Kebutuhan Kinerja  
Kebutuhan Infrastruktur  
Kebutuhan Pemeliharaan  
Kebutuhan Manajemen Data



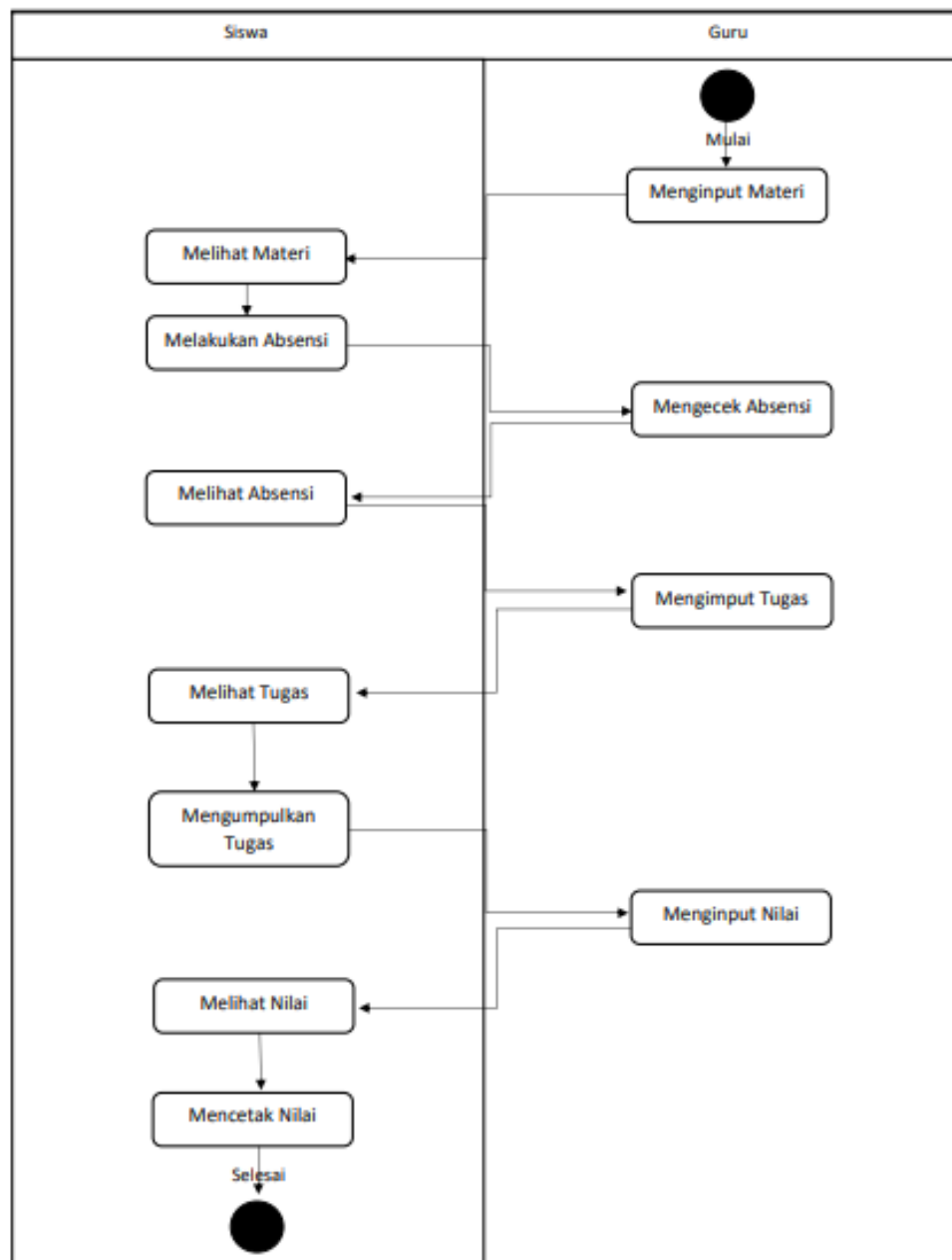
c. Gambarkan usecase diagramnya!

Usecase Diagram :



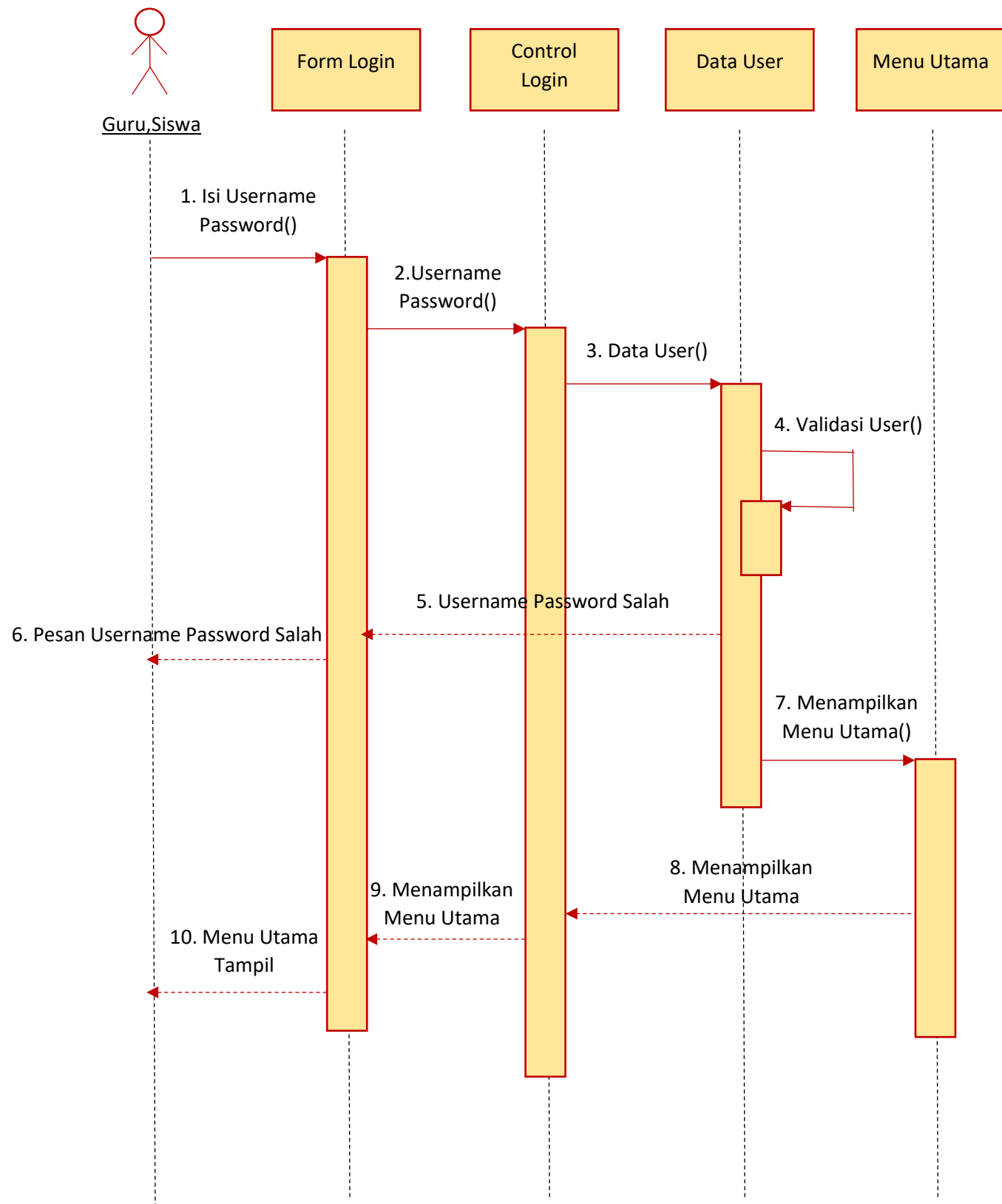
d. Gambarkan activity diagramnya!

Activity Diagram :

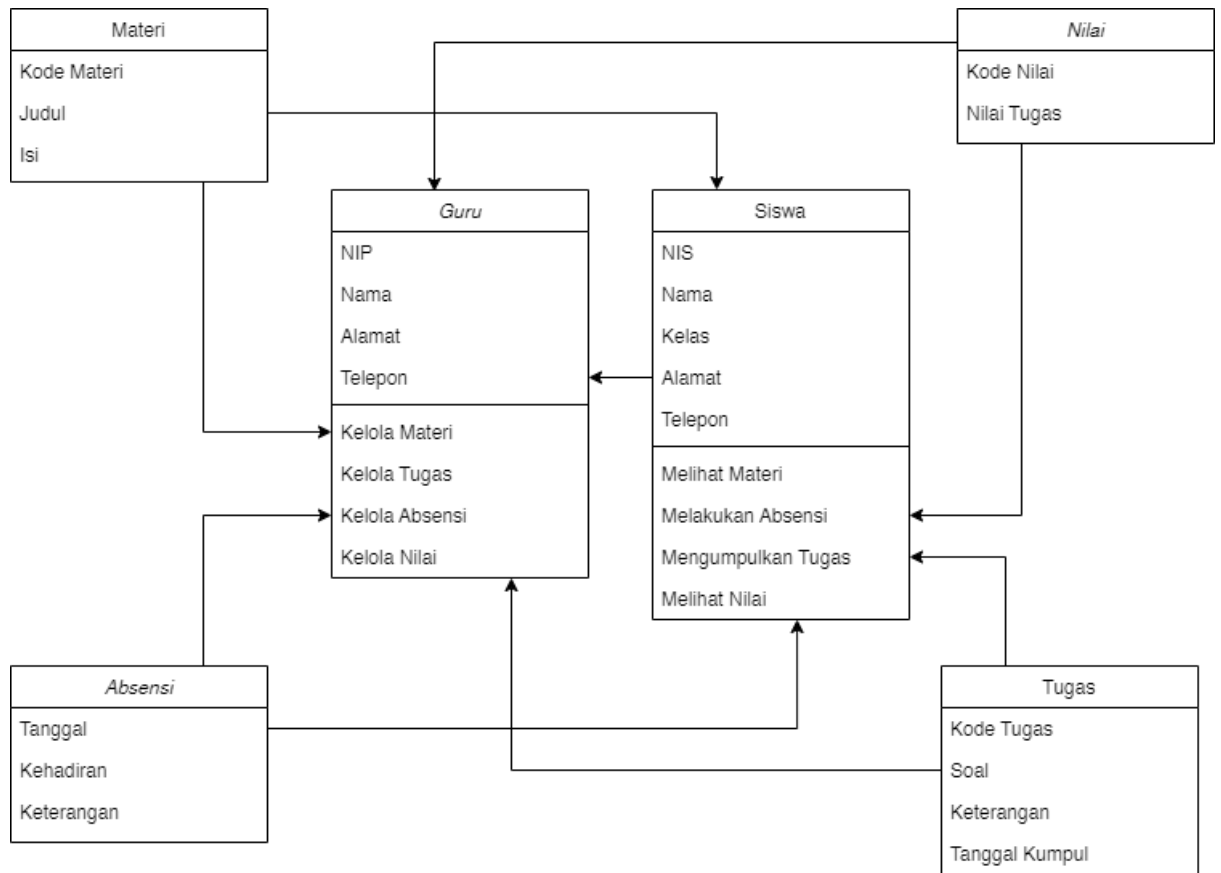


e. Gambarkan sequence diagramnya!

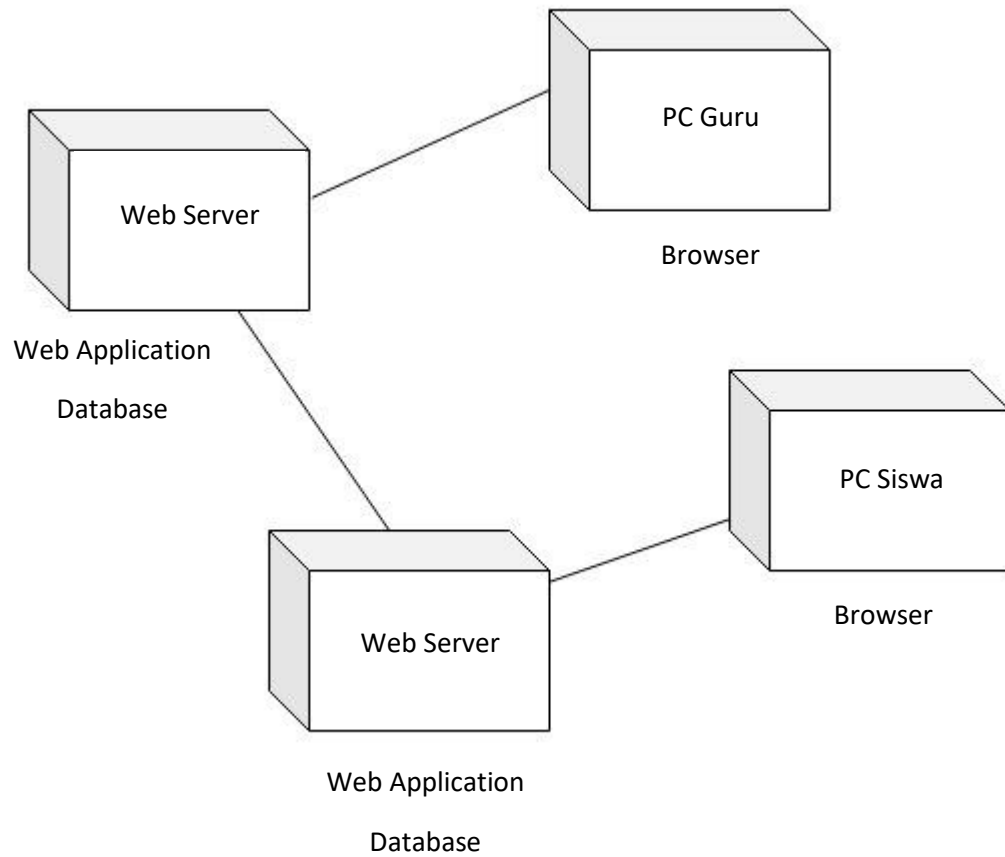
Sequence Diagram :



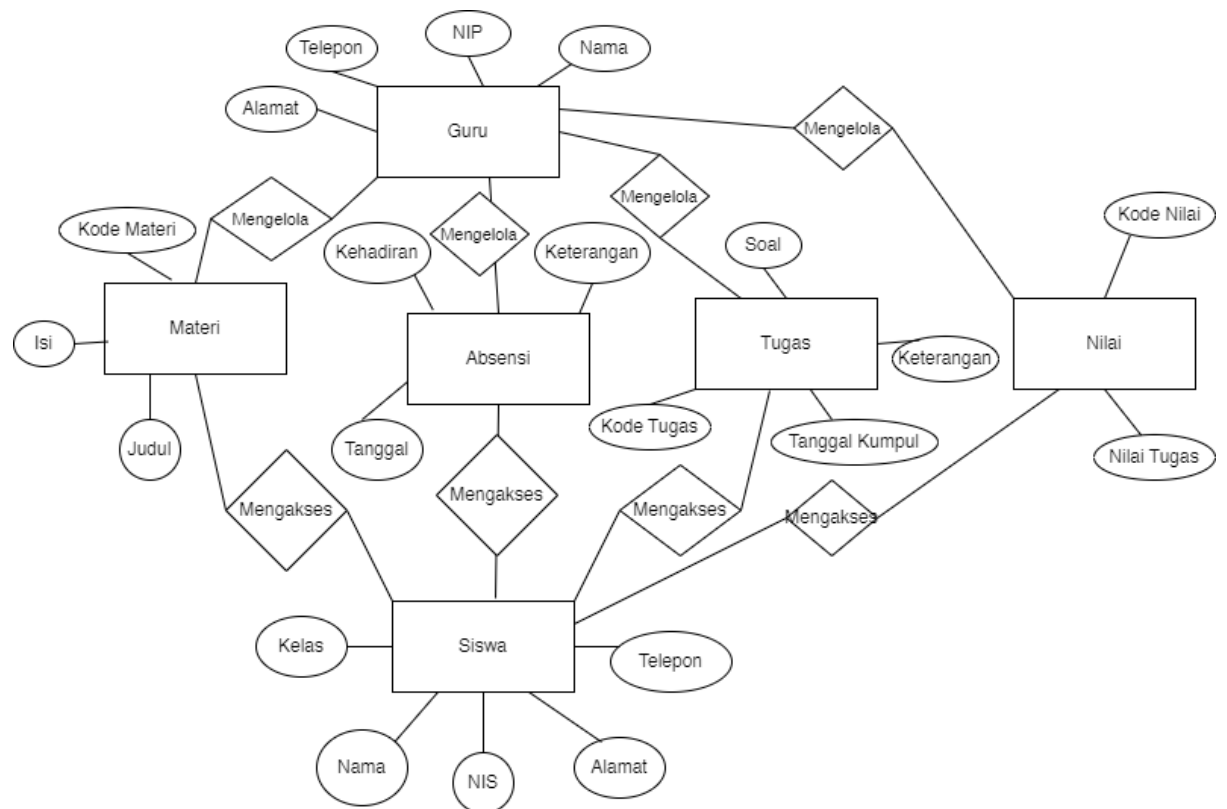
8. Selanjutnya adalah tahap perancangan atau desain:  
a. Gambarkan Class Diagramnya!



b. Gambarkan Deployment Diagramnya!

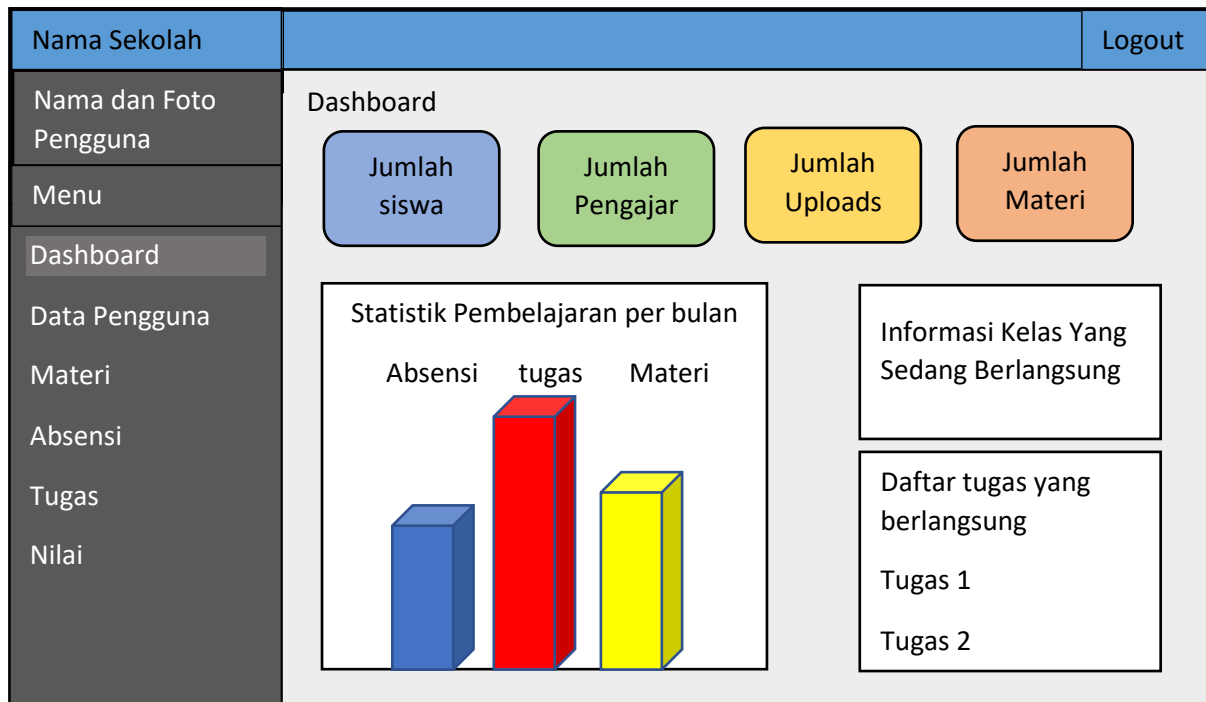


c. Gambarkan Data Model Diagramnya!



d. Gambarkan User Interface!

### Dashboard



## Data Pengguna

Nama Sekolah			Logout
Nama dan Foto Pengguna	Data Pengguna		
Menu	<div></div> <p>Data-data pengguna yang dapat di edit</p>		
Dashboard			
Data Pengguna			
Materi			
Absensi			
Tugas			
Nilai			



## Materi

Nama Sekolah	Logout	
Nama dan Foto Pengguna	Materi	
Menu	<div>Tahun PelajaranKelasPilih mata pelajaran</div>	
Dashboard	<div>Judul Materi</div> <div>Isi Materi</div>	
Data Pengguna		
Materi		
Absensi		
Tugas		
Nilai		

## Absensi

Nama Sekolah			Logout
Nama dan Foto Pengguna	Absensi		
Menu	Tahun Pelajaran	Kelas	Pilih mata pelajaran
Dashboard	Info kehadiran per pertemuan 1. hadir 2. hadir 3. sakit 4. hadir 5. tanpa keterangan 6. hadir		Pertemuan ke 4
Data Pengguna			hadir
Materi			Sakit   Izin
Absensi			Keterangan
Tugas			Submit
Nilai			

## Tugas

Nama Sekolah			Logout
Nama dan Foto Pengguna	Tugas		
Menu	Tahun Pelajaran	Kelas	Pilih mata pelajaran
Dashboard	Durasi Pengerjaan Tugas	Tugas Ke 3	
Data Pengguna	Isi tugas	Masukkan file tugas	
Materi		Daftar File tugas yang ditambahkan	
Absensi		Kirim	
Tugas			
Nilai			

## Nilai

Nama Sekolah			Logout																														
Nama dan Foto Pengguna	Nilai																																
Menu	<div>Tahun Pelajaran</div> <div>Kelas</div>																																
Dashboard	<div>Daftar Nilai Semua mata pelajaran</div> <table><thead><tr><th>Mapel</th><th>Tugas 1</th><th>Tugas 2</th><th>Tugas 3</th><th>Tugas 4</th><th>Total nilai</th></tr></thead><tbody><tr><td>Mapel 1</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Mapel 2</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Mapel 3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Mapel 4</td><td></td><td></td><td></td><td></td><td></td></tr></tbody></table>			Mapel	Tugas 1	Tugas 2	Tugas 3	Tugas 4	Total nilai	Mapel 1						Mapel 2						Mapel 3						Mapel 4					
Mapel				Tugas 1	Tugas 2	Tugas 3	Tugas 4	Total nilai																									
Mapel 1																																	
Mapel 2																																	
Mapel 3																																	
Mapel 4																																	
Data Pengguna																																	
Materi																																	
Absensi																																	
Tugas																																	
Nilai																																	