

PERTEMUAN 5

24/09/2024

Unified Modeling Language (UML) adalah bahasa pemodelan standar yang digunakan dalam **pengembangan perangkat lunak** untuk membantu desainer dan pengembang sistem dalam menggambarkan, merancang, dan mendokumentasikan sistem perangkat lunak. UML tidak hanya digunakan dalam pengembangan perangkat lunak, tetapi juga bisa diterapkan untuk memodelkan sistem bisnis, proses organisasi, dan bahkan proses non-teknis.

Tujuan UML:

- **Visualisasi:** Membantu menggambarkan sistem perangkat lunak secara visual.
- **Spesifikasi:** Menggambarkan rincian struktur dan perilaku sistem.
- **Konstruksi:** Menyediakan panduan selama implementasi sistem.
- **Dokumentasi:** Menghasilkan dokumentasi yang lengkap dan jelas mengenai sistem.

Komponen Utama dalam UML:

1. **Structural Diagrams (Diagram Struktural):** Digunakan untuk menggambarkan struktur statis sistem, yaitu bagaimana elemen-elemen dalam sistem diorganisir.
 - **Class Diagram:** Menggambarkan kelas-kelas dalam sistem dan hubungan antar kelas tersebut.
 - **Object Diagram:** Menggambarkan objek dalam sistem dan hubungan antar objek.
 - **Component Diagram:** Menjelaskan hubungan antar komponen dalam sistem.
 - **Deployment Diagram:** Menunjukkan distribusi fisik dari sistem pada hardware.
 - **Package Diagram:** Menggambarkan pengelompokan elemen-elemen UML ke dalam paket.
 - **Composite Structure Diagram:** Menjelaskan struktur internal dari kelas atau komponen.
2. **Behavioral Diagrams (Diagram Perilaku):** Digunakan untuk memodelkan perilaku dinamis sistem, yaitu bagaimana sistem bereaksi terhadap input dan bagaimana sistem merespons berbagai tindakan.
 - **Use Case Diagram:** Menggambarkan interaksi antara aktor eksternal (pengguna) dengan sistem dan fungsi utama sistem.
 - **Sequence Diagram:** Menggambarkan interaksi objek dalam urutan waktu tertentu.
 - **Activity Diagram:** Menggambarkan aliran kontrol dan alur aktivitas dalam sistem.

- **State Machine Diagram:** Menjelaskan transisi antara status-status objek dalam sistem.
 - **Communication Diagram:** Menggambarkan interaksi antara objek dengan menyoroti pesan yang dikirimkan.
 - **Interaction Overview Diagram:** Menyediakan gambaran umum tentang bagaimana interaksi dalam sistem berlangsung.
3. **Interaction Diagrams (Diagram Interaksi):** Diagram ini lebih fokus pada aliran komunikasi antara objek-objek.
- **Sequence Diagram:** Menggambarkan urutan pesan yang dipertukarkan antar objek.
 - **Communication Diagram:** Berfokus pada interaksi pesan antar objek.

Elemen Utama dalam UML:

1. **Aktor:** Representasi dari pengguna atau sistem lain yang berinteraksi dengan sistem.
2. **Kelas:** Kelas-kelas dalam sistem, masing-masing memiliki atribut dan metode.
3. **Objek:** Contoh dari kelas yang digunakan untuk merepresentasikan elemen nyata dalam sistem.
4. **Relasi:** Hubungan antara kelas atau objek, seperti asosiasi, agregasi, dan komposisi.
5. **Skenario:** Serangkaian interaksi atau langkah yang menggambarkan bagaimana pengguna berinteraksi dengan sistem.

Contoh Penggunaan UML:

- **Use Case Diagram** untuk menggambarkan bagaimana pengguna berinteraksi dengan sistem.
- **Class Diagram** untuk mendesain struktur kelas dalam suatu perangkat lunak.
- **Sequence Diagram** untuk menggambarkan alur proses dari suatu fungsi atau operasi di dalam sistem.
- **Activity Diagram** untuk memodelkan alur kerja (workflow) dari proses bisnis atau proses dalam sistem.

Penerapan UML dalam Industri:

UML sering digunakan dalam berbagai sektor bisnis, seperti:

- **Pengembangan perangkat lunak:** Sebagai standar de facto dalam mendokumentasikan sistem perangkat lunak, termasuk di perusahaan teknologi.
- **Perbankan dan keuangan:** Untuk memodelkan proses bisnis kompleks, seperti alur transaksi dan manajemen risiko.
- **Industri manufaktur:** Untuk memodelkan proses produksi dan operasi, serta mengelola siklus hidup produk.

Kelebihan UML:

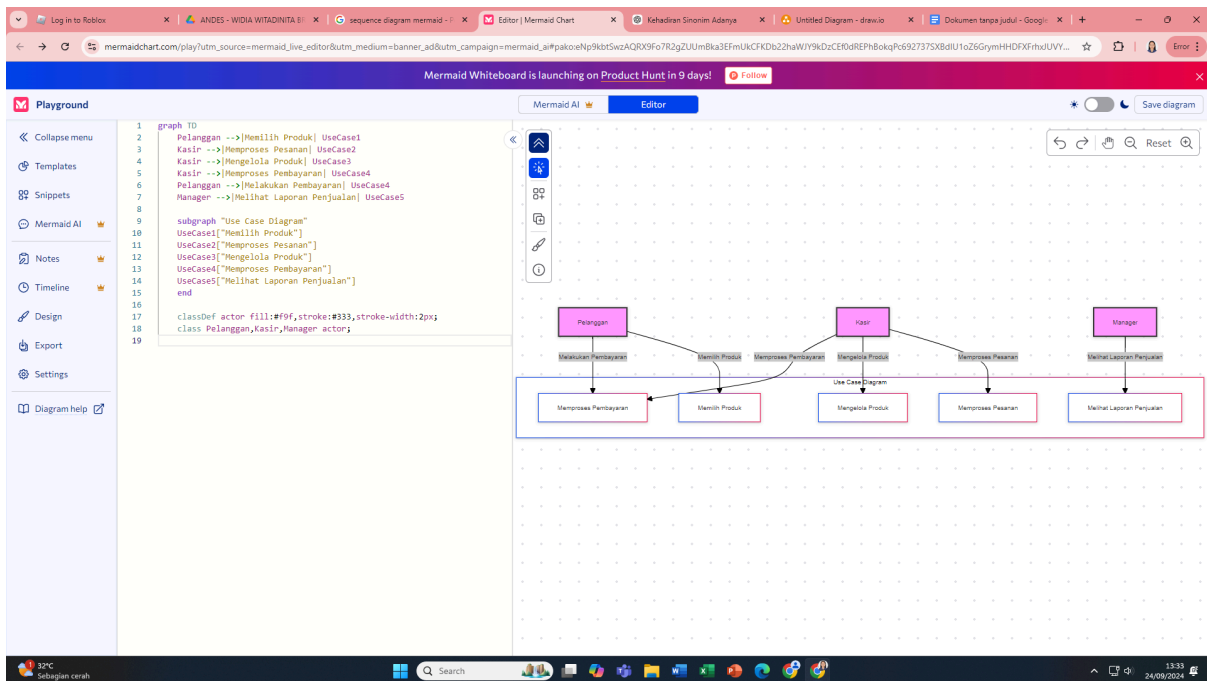
- **Standarisasi:** UML diterima secara luas sebagai standar dalam pemodelan sistem perangkat lunak.
- **Fleksibilitas:** Dapat digunakan untuk memodelkan sistem yang kompleks di berbagai sektor industri.
- **Visualisasi yang baik:** Membantu para pengembang, desainer, dan pemangku kepentingan lain memahami dan berkomunikasi dengan lebih baik.

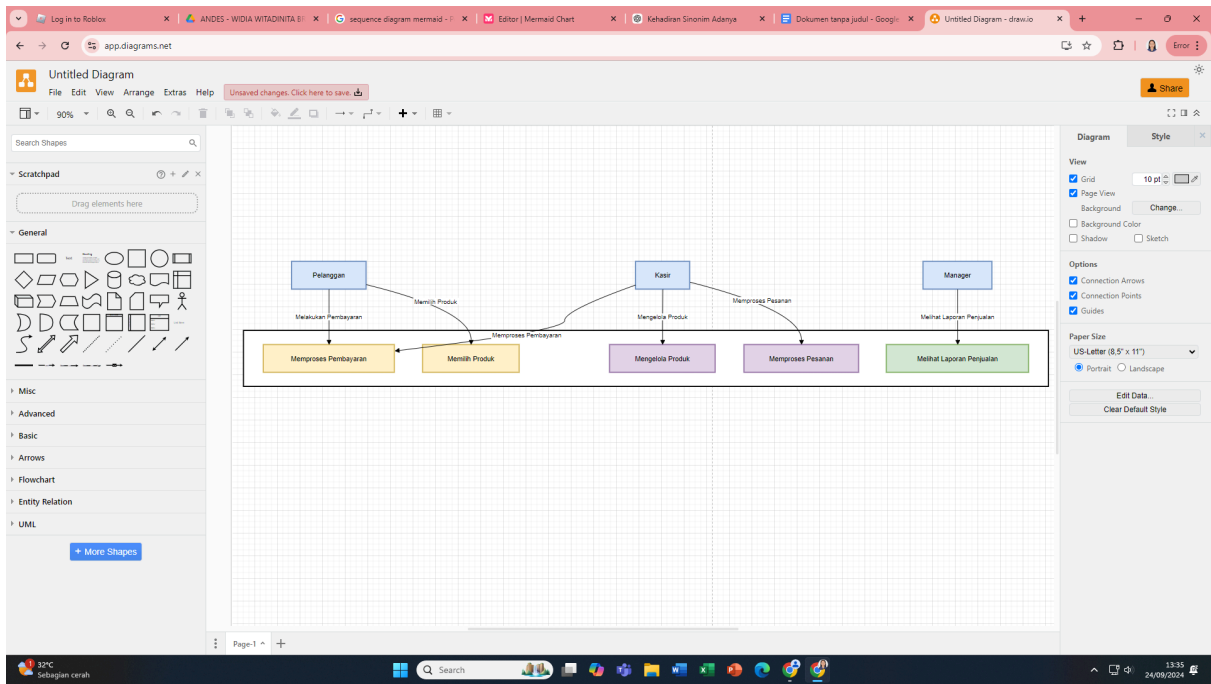
Kesimpulan:

UML adalah bahasa pemodelan yang penting untuk mendesain dan mendokumentasikan sistem perangkat lunak yang kompleks. Dengan diagram-diagramnya yang berbeda, UML memungkinkan pengembang untuk memvisualisasikan arsitektur sistem secara rinci dan efektif.

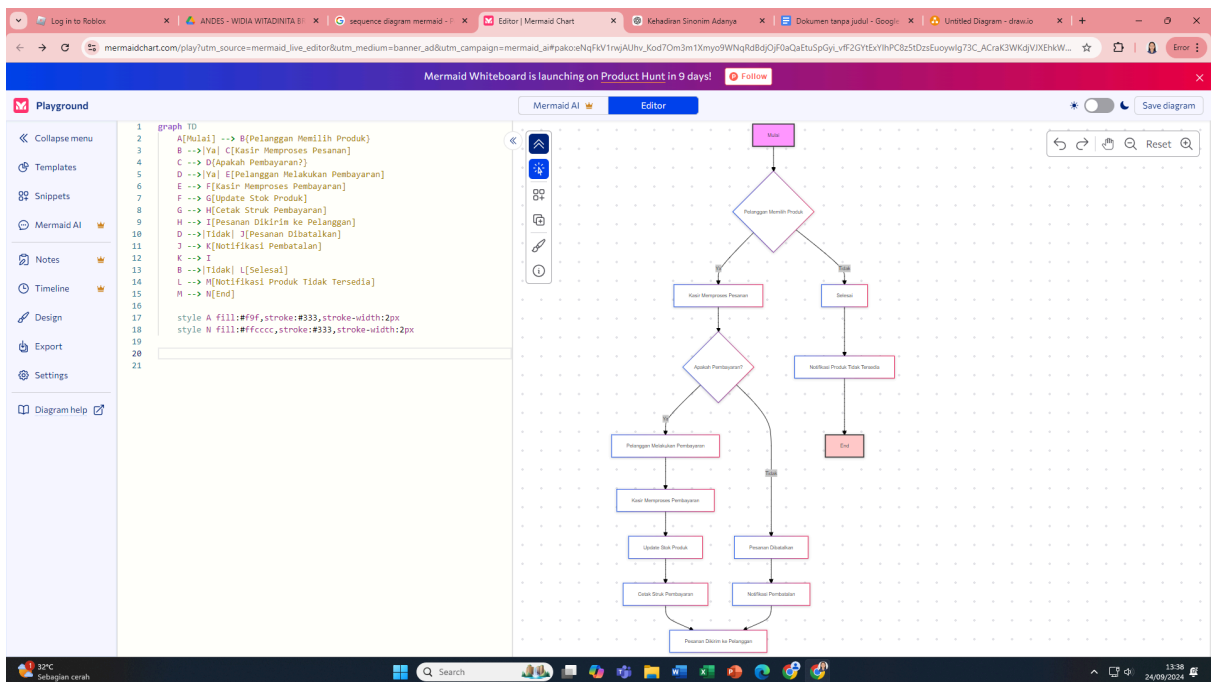
Referensi UML digunakan untuk memastikan desain sistem yang komprehensif dan terdokumentasi dengan baik sehingga sistem dapat dibangun dan dikembangkan dengan efisien.

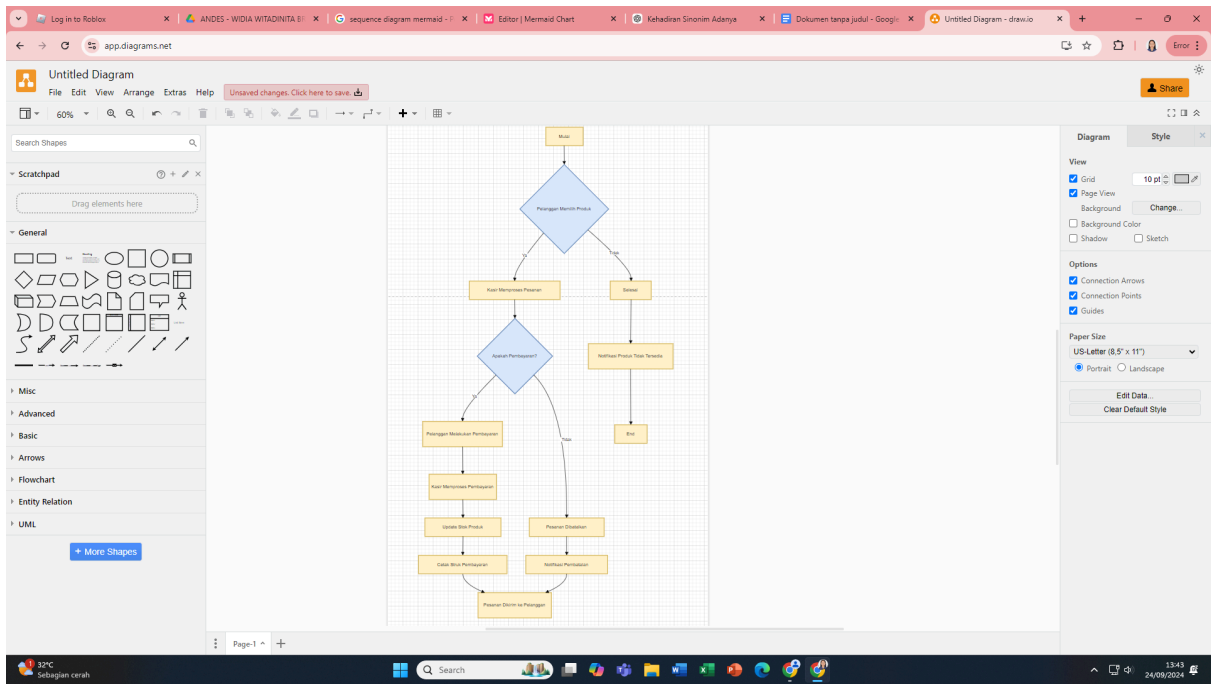
1. Use Case Diagram



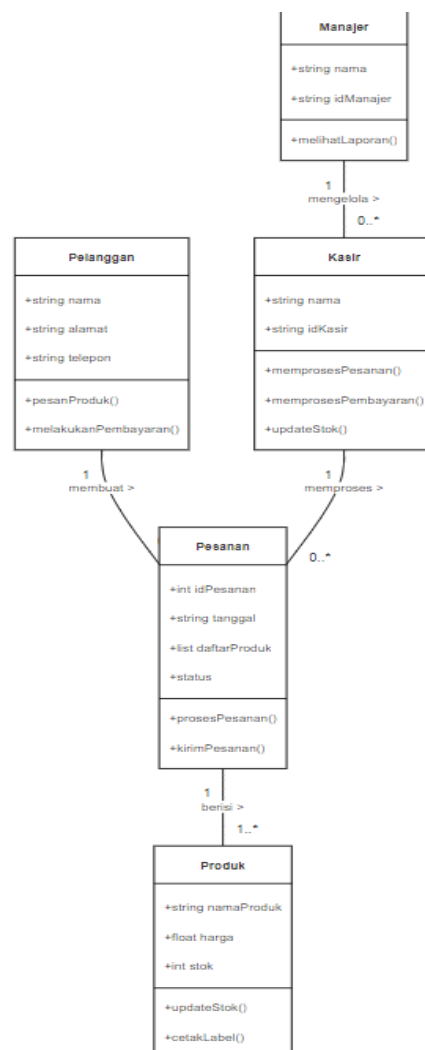


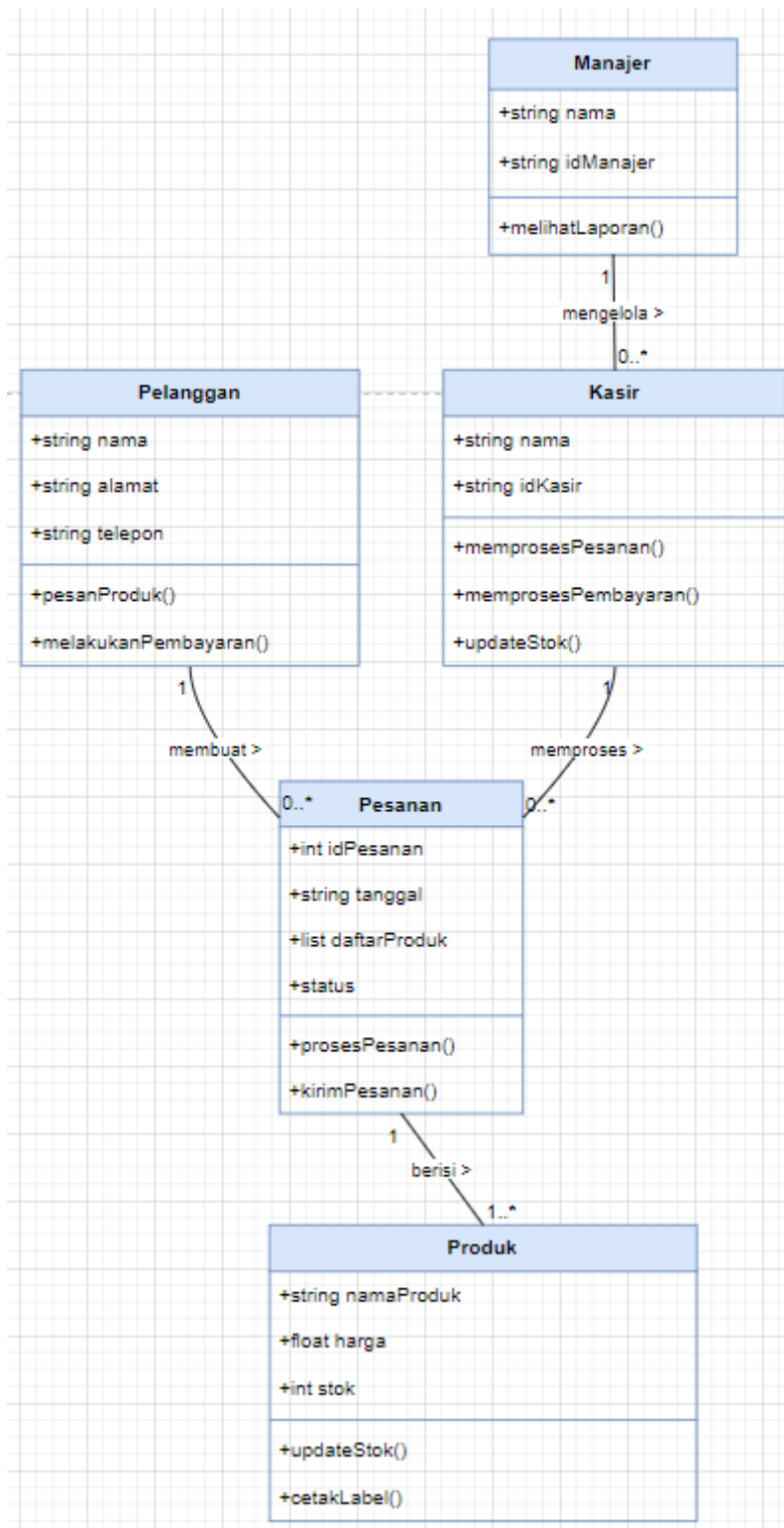
2. Activity Diagram



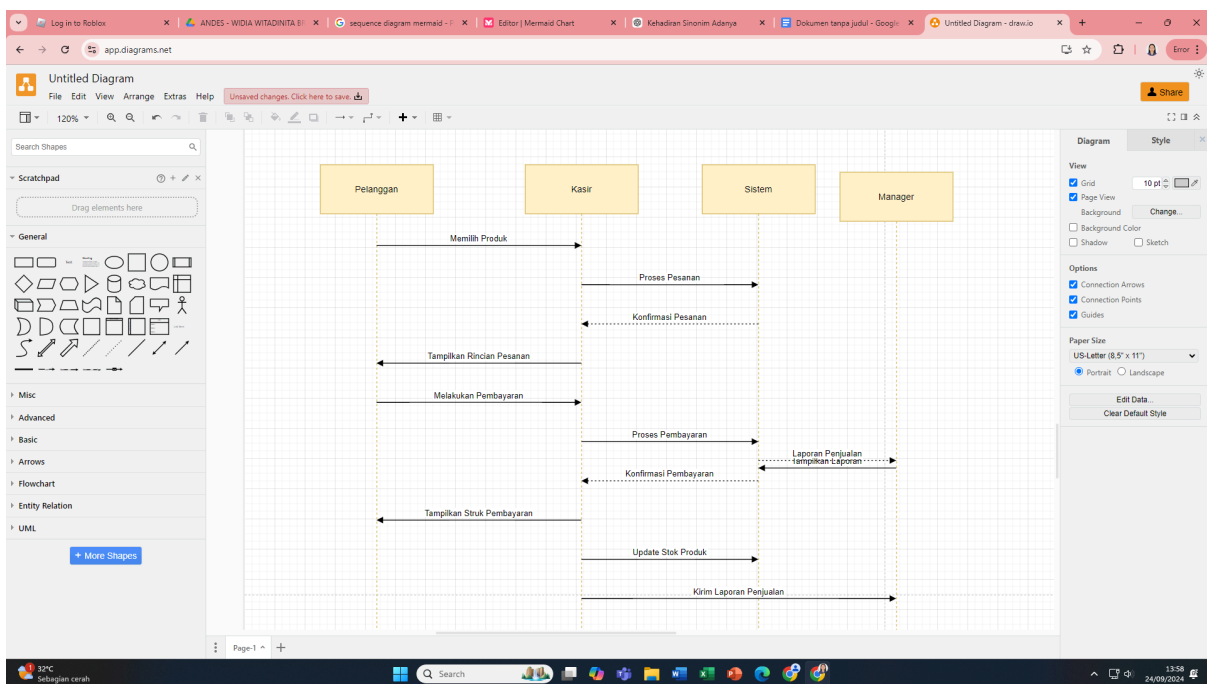
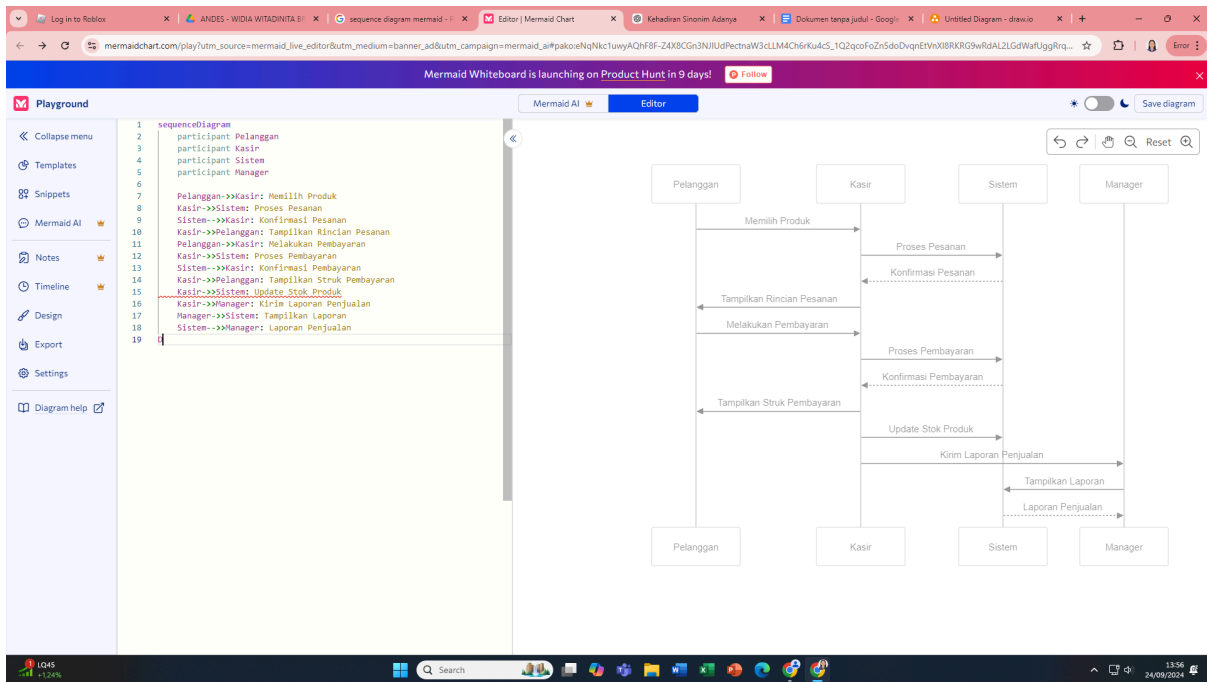


3. Class Diagram





4. Sequence Diagram



5. State Diagram

