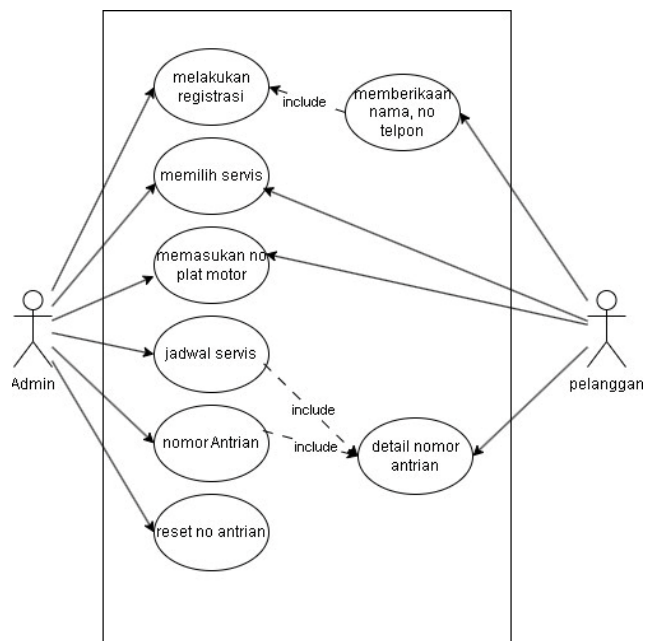


## APLIKASI DAN DIAGRAM UML

### A. Deskripsi Aplikasi AMBAGARAGE

Aplikasi AMBAGARAGE adalah sistem pemesanan layanan bengkel online berbasis *web* yang memungkinkan pengguna memesan servis seperti *Tune Up*, Servis Ringan, Servis Berat, *Full Servis*, dan Ganti Oli. Pengguna dapat memilih layanan, mengisi data (nama, nomor telepon, plat kendaraan), dan mendapatkan nomor antrian beserta konfirmasi pemesanan. Aplikasi ini juga memiliki fitur reset antrian untuk memulai ulang nomor antrian.

### B. Use Case



Gambar 1 Use Case

Penjelasan :

#### ➤ Pelanggan

- Melakukan registrasi: Pelanggan harus mendaftarkan diri dengan memberikan informasi seperti nama dan nomor telepon (use case "memberikan nama, no telpon").
- Memilih servis: Pelanggan dapat memilih jenis layanan seperti Tune Up, Servis Ringan, dll.

- Memasukkan nomor plat motor: Setelah memilih layanan, pelanggan harus memasukkan nomor plat kendaraan mereka.
- Detail nomor antrean: Setelah selesai, pelanggan akan mendapatkan nomor antrean yang berisi informasi lengkap tentang pesanan.

➤ **Admin**

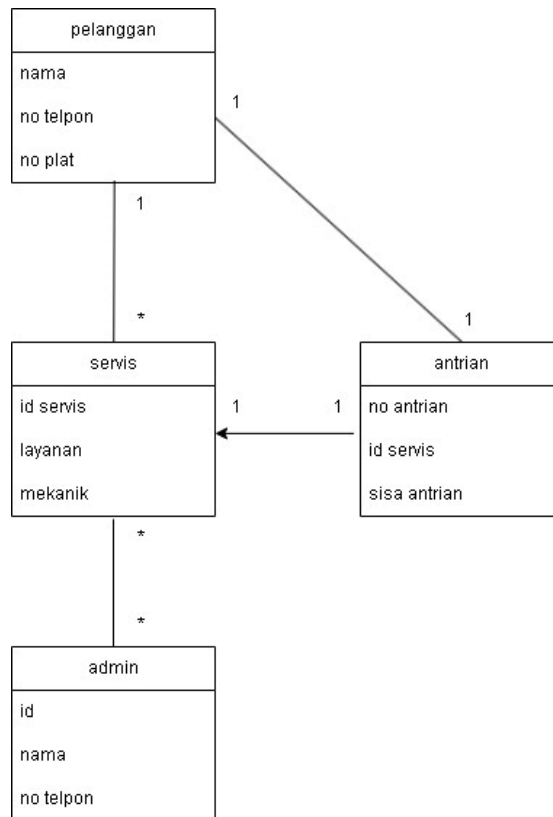
- Mengatur jadwal servis: Admin bertugas untuk mengelola jadwal layanan berdasarkan antrean yang telah dipesan oleh pelanggan.
- Nomor antrean: Admin juga dapat melihat nomor antrean yang telah dibuat dan mengelolanya.
- Reset nomor antrean: Admin memiliki fitur untuk mereset nomor antrean jika diperlukan (misalnya, memulai antrean dari awal).

➤ **Interaksi:**

- Pelanggan adalah pengguna utama yang berinteraksi dengan aplikasi untuk melakukan pemesanan.
- Admin bertugas untuk memonitor dan mengelola data yang diinput pelanggan.

Diagram ini menunjukkan bagaimana sistem mengatur interaksi antara pelanggan dan admin dalam proses pemesanan layanan di aplikasi AMBAGARAGE.

### C. Class Diagram



Gambar 2 Class Diagram

Penjelasan :

#### 1. Kelas Pelanggan:

##### ➤ Atribut:

- nama: Nama pelanggan.
- no\_telpon: Nomor telepon pelanggan.
- no\_plat: Nomor plat kendaraan pelanggan.

##### Hubungan:

Setiap pelanggan memiliki hubungan 1 ke banyak dengan servis (pelanggan dapat memesan lebih dari satu layanan).

#### 2. Kelas Servis:

##### ➤ Atribut:

- id\_servis: Identifikasi unik untuk setiap jenis servis.
- layanan: Jenis layanan servis yang tersedia (contoh: Tune Up, Ganti Oli, dll.).

- mekanik: Informasi mengenai mekanik yang menangani servis.

**Hubungan:**

- Setiap servis berhubungan dengan satu pelanggan tetapi bisa memiliki banyak antrean.
- Admin dapat mengelola berbagai layanan servis (hubungan \* ke banyak dengan admin).

**3. Kelas Antrean:**

➤ **Atribut:**

- no\_antrian: Nomor antrean yang diberikan kepada pelanggan.
- id\_servis: Referensi ke jenis layanan yang dipilih.
- sisa\_antrian: Menyimpan informasi mengenai sisa antrean yang belum selesai.

**Hubungan:**

Antrean berhubungan 1 ke 1 dengan servis.

**4. Kelas Admin:**

➤ **Atribut:**

- id: Identifikasi unik untuk setiap admin.
- nama: Nama admin.
- no\_telpon: Nomor telepon admin.

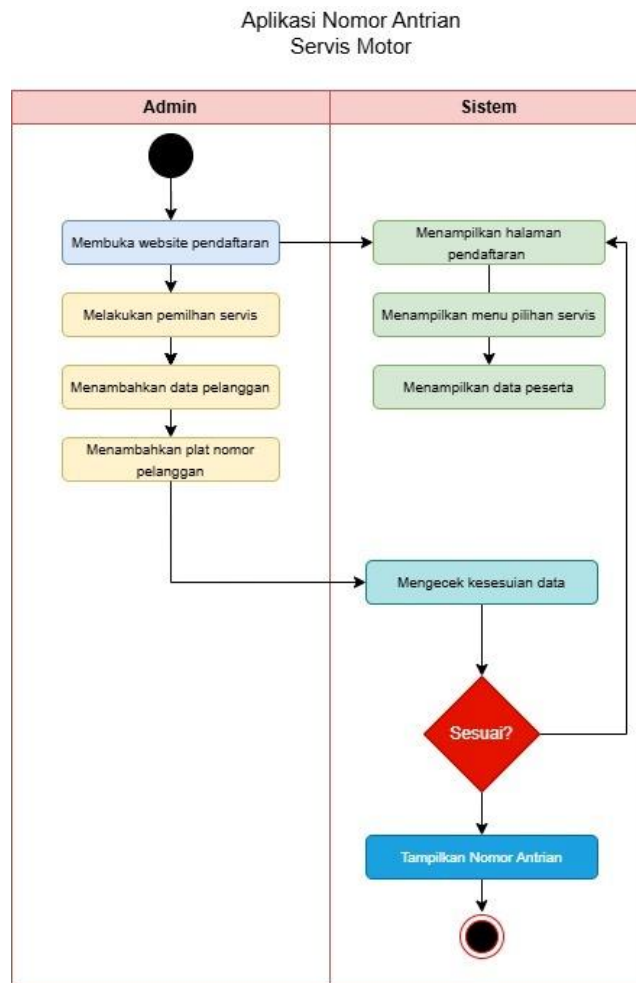
**Hubungan:**

Admin memiliki hubungan 1 ke banyak dengan servis (admin dapat mengelola banyak layanan).

**Hubungan Utama:**

- Pelanggan melakukan pesanan untuk layanan pada Servis, yang kemudian menghasilkan Antrean.
- Admin bertugas mengelola berbagai layanan pada Servis untuk memproses antrean pelanggan.

#### D. Activity Diagram



Gambar 3 Activity Diagram

Penjelasan:

##### Pihak Terlibat:

1. Admin: Bertugas memasukkan data pelanggan dan melakukan pendaftaran layanan.
2. Sistem: Memproses data yang dimasukkan admin dan menampilkan hasilnya.

##### Alur Aktivitas:

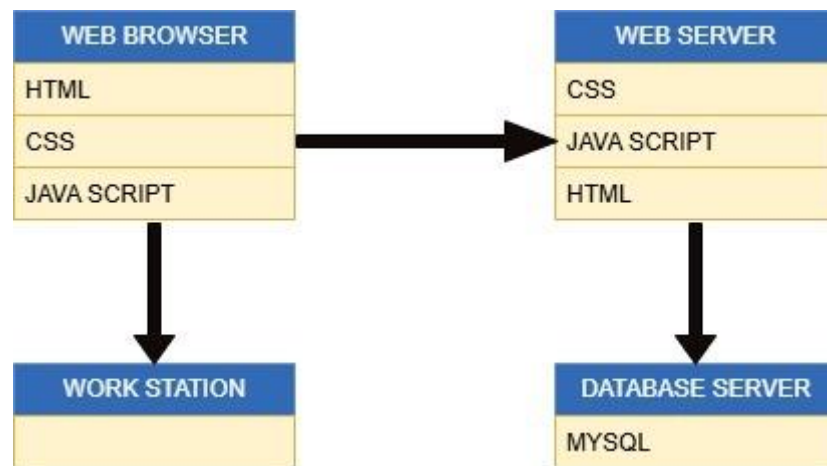
1. Admin Membuka Website Pendaftaran:
  - Admin memulai proses dengan mengakses aplikasi melalui website.
2. Sistem Menampilkan Halaman Pendaftaran:

- Sistem memuat halaman pendaftaran yang berisi menu dan opsi layanan.
3. Admin Melakukan Pemilihan Servis:
    - Admin memilih jenis layanan yang diinginkan pelanggan, seperti Tune Up atau Ganti Oli.
  4. Sistem Menampilkan Menu Pilihan Servis:
    - Sistem menunjukkan daftar pilihan servis yang tersedia.
  5. Admin Menambahkan Data Pelanggan:
    - Admin mengisi data pelanggan, seperti nama, nomor telepon, dan nomor plat kendaraan.
  6. Sistem Menampilkan Data Peserta:
    - Sistem menunjukkan data pelanggan yang sudah diinput untuk diperiksa.
  7. Sistem Mengecek Kesesuaian Data:
    - Sistem memverifikasi apakah data yang dimasukkan sesuai dengan format dan kebutuhan.
  8. Keputusan Kesesuaian Data:
    - Jika data sesuai, sistem akan melanjutkan ke proses berikutnya.
    - Jika data tidak sesuai, admin harus memperbaiki data yang diinput.
  9. Menampilkan Nomor Antrean:
    - Setelah data sesuai, sistem menghasilkan nomor antrean dan menampilkannya kepada admin untuk disampaikan ke pelanggan.

**Tujuan:**

Diagram ini menunjukkan alur bagaimana admin dan sistem bekerja sama untuk mendaftarkan pelanggan ke layanan bengkel dengan menghasilkan nomor antrean yang valid. Proses ini memastikan data pelanggan akurat dan layanan dapat dikelola dengan baik.

### E. Deployemnt Diagram



Gambar 4 Deployemnt Diagram

Penjelasan:

#### 1. Web Browser:

- Komponen ini berada di sisi pengguna (*client*) dan digunakan untuk mengakses aplikasi melalui internet.
- Pembuatan aplikasi meliputi:
  - HTML: Untuk struktur halaman web.
  - CSS: Untuk gaya tampilan antarmuka.
  - JavaScript: Untuk interaktivitas di sisi pengguna.

#### 2. Web Server:

- Komponen server yang mengelola permintaan dari pengguna melalui web browser.
- Menggunakan teknologi yang sama (HTML, CSS, JavaScript) untuk memproses dan mengirimkan data ke browser.
- Berfungsi sebagai penghubung antara browser dan database server.

#### 3. Database Server:

- Komponen ini menyimpan semua data yang diperlukan oleh aplikasi.
- Menggunakan MySQL sebagai sistem manajemen basis data untuk menyimpan, mengelola, dan menyediakan data saat diperlukan.

#### 4. Workstation:

- Digunakan sebagai perangkat kerja pengguna untuk mengakses web browser dan menjalankan aplikasi.
- Berfungsi sebagai medium fisik untuk mengoperasikan aplikasi.

Alur Kerja:

- Pengguna mengakses aplikasi melalui *web browser* di workstation.
- Permintaan dikirimkan ke *web server*, di mana data diolah dan respon dikembalikan ke browser.
- Jika data yang diminta melibatkan penyimpanan atau pengambilan informasi, *web server* akan berinteraksi dengan *database server* untuk memperoleh atau menyimpan data.

## F. Skenario Bisnis AMBAGARAGE

### 1. Skenario Pemesanan Layanan Servis oleh Pelanggan

Pelanggan:

Seorang pelanggan, malikus, ingin melakukan servis *Tune Up* pada kendaraannya. Namun, ia tidak ingin menghabiskan waktu lama di bengkel untuk mengantri.

**Langkah-langkah:**

- 1) Malikus membuka aplikasi AMBAGARAGE melalui web browser di laptopnya.
- 2) Ia memilih menu layanan yang diinginkan, yaitu "*Tune Up*".
- 3) Malikus memasukkan data seperti nama, nomor telepon, dan nomor plat kendaraan.
- 4) Sistem memproses data yang dimasukkan dan memberikan nomor antrean kepada Malikus.
- 5) Malikus mendapatkan detail pemesanan (nomor antrean, jenis layanan, estimasi waktu pengerjaan).

**Hasil:**

Malikus datang ke bengkel sesuai jadwal yang telah ditentukan tanpa harus menunggu lama di tempat.

### 2. Skenario Pengelolaan Antrean oleh Admin



Admin Bengkel:

Andi, admin bengkel, ingin mengelola antrean yang sudah dipesan oleh pelanggan secara terorganisir.

**Langkah-langkah:**

- 1) Andi membuka aplikasi AMBAGARAGE melalui web browser di komputer kerja.
- 2) Ia melihat daftar antrean pelanggan yang sudah masuk, lengkap dengan data layanan yang mereka pilih.
- 3) Jika ada pelanggan yang membatalkan antreannya, Andi dapat menggunakan fitur "Reset Antrean" untuk memperbarui antrean.
- 4) Andi juga dapat memeriksa jadwal layanan untuk memastikan efisiensi operasional bengkel.

**Hasil:**

Andi dapat mengelola antrean pelanggan dengan baik sehingga operasional bengkel berjalan lancar.

3. Skenario Penyimpanan Data dan Pengelolaan Riwayat Pelanggan

Manajer Bengkel:

Pak Arif, manajer bengkel, ingin mengevaluasi jumlah pelanggan yang menggunakan layanan Tune Up dalam sebulan terakhir.

**Langkah-langkah:**

- 1) Pak Arif mengakses aplikasi AMBAGARAGE untuk melihat data pelanggan yang tersimpan di database (di-backend menggunakan MySQL).
- 2) Ia mengekspor laporan dari sistem yang berisi daftar pelanggan beserta jenis layanan yang mereka gunakan.
- 3) Dengan data ini, Pak Arif menganalisis popularitas setiap jenis layanan dan merencanakan promosi di bulan depan.

**Hasil:**

Pak Arif memiliki wawasan yang lebih baik tentang kebutuhan pelanggan dan dapat meningkatkan strategi pemasaran bengkel.