

Nama Artefak	:	Sistem Parkir DJ Ongkok <i>Restaurant</i>
Dikerjakan Oleh	:	12S18002 – Wiranda Siahaan 12S18005 – Lusiana Siahaan 12S18062 – Ester Putri Sidabutar 12S18067 – Grace Panjaitan

1. Ilustrasi DJ Ongkok *Restaurant*

a. Topik Permasalahan

Bisnis restoran semakin digemari pengusaha muda milenial. Seorang DJ (*Disc Jokey*) terkenal ikut terjun dalam bisnis ini dan membuka sebuah *restaurant* dengan konsep ala-ala Jamaika. Restoran itu dinamai DJ Ongkok *Restaurant* (DJOR). DJOR dilengkapi dengan area nongkrong dan parkir yang lumayan luas, hal ini menjadi daya tarik tersendiri selain makanannya yang tak seberapa itu. DJOR dikenal dengan *tagline* “rasa kaki lima harga bintang lima”. Sudah beberapa bulan buka, DJOR yang berlokasi di tepi jalan kota yang ramai mulai mendapatkan sebuah persoalan terkait parkir. Pada jam-jam tertentu, misal waktu makan siang dan makan malam, banyak pelanggan yang datang dengan membawa kendaraan pribadinya. Persoalannya, pelanggan mengeluh sulit menemukan lokasi parkir dan akibatnya timbul antrian yang tak jarang menimbulkan kemacetan di jalan kota. Wajar saja, DJOR sangat ramai dan digandrungi oleh anak-anak muda. Salah satu solusi yang dapat dilakukan adalah dengan dengan menginformasikan lokasi-lokasi parkir yang kosong kepada pelanggan yang datang sehingga menurunkan perlambatan kendaraan yang akan parkir. Dimana pada system akan di tampilkan lokasi parkir dengan menggunakan 2 jenis warna yaitu:

1. Warna hijau yang menyatakan lokasi parkir tersebut sedang kosong.
2. Waran merah yang menyatakan lokasi parkir tersebut sedang tidak kosong.

Setiap kali masuk, pelanggan akan memilih pada lokasi mana ia akan parkir, kumpulan informasi ini kemudian digunakan oleh sistem untuk menginformasikan lokasi mana yang masih kosong pada pelanggan berikutnya. Perlu diketahui, area parkir DJOR terdiri atas empat *sections* (*Section A, B, C, dan D*), yang setiap *section* mampu menampung enam buah mobil. Keuntungan lain yang dapat diperoleh dari menerapkan sistem parkir ini adalah

penghitungan dan pencatatan biaya parkir dapat dilakukan secara otomatis. Untuk satu jam pertama tidak dikenai biaya, selanjutnya dikenai tarif Rp20.000 per jam.

b. Ilustrasi Sistem Parkir DJOR

Sebelum melakukan parkir, pengemudi akan membuka aplikasi terlebih dahulu. Kemudian aplikasi akan menunjukkan denah parkir dengan contoh gambar sebagai berikut.



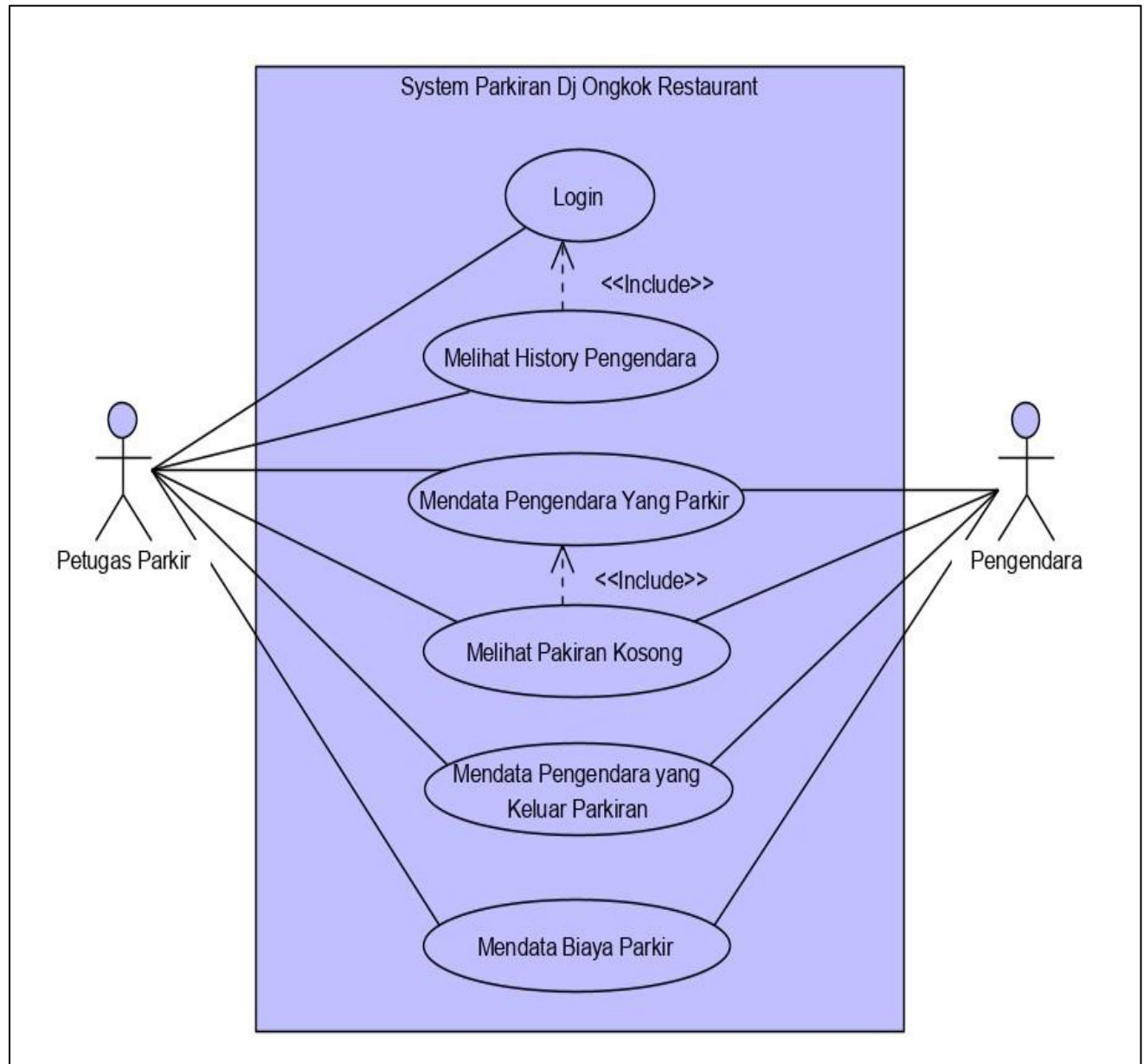
Setiap slot diberi nomor agar pengendara yang akan memarkir agar mengurangi resiko kesalahan memarkir. Slot yang diberi warna merah menandakan slot sudah ditempati. Slot yang diberi warna hijau menandakan slot masih kosong.

Pengendara yang akan memarkir menekan slot berwarna hijau. Setelah itu, pengendara mengisi informasi yang ditampilkan oleh sistem. Setelah itu, informasi di simpan ke dalam sistem dan slot berubah warna menjadi merah dan menampilkan nomor plat kendaraan. Setelah itu, sistem secara otomatis menghitung waktu parkir. Sistem juga akan memberikan *kode Verifikasi* kepada pengendara yang sedang memarkir.

Pengendara yang akan meninggalkan tempat parkir terlebih dahulu menemui petugas parkir di portal keluar. Pengendara kemudian memberikan *kode Verifikasi* untuk melakukan verifikasi/validasi. Setelah verifikasi melalui sistem, sistem akan menampilkan biaya parkir pengendara.

Sistem juga dapat memperlihatkan jumlah pengendara yang sudah parkir pada waktu tertentu. Tujuan dibuatnya fitur seperti ini pada sistem adalah sebagai referensi yang dapat membantu pemilik *restaurant* menentukan apakah tempat parkir perlu diekspansi atau tidak.

2. Use case diagram



Gambar 1. Use case diagram Sistem Parkiran Dj Ongkok Restaurant

3. Use case scenario

Tabel 1. [UCS-01]- Log-in

Identifier	:	UC01
Goal	:	Petugas parkir berhasil melakukan log-in kedalam sistem .
Primary actor	:	Petugas parkir.
Secondary actor	:	-
Trigger	:	Petugas parkir ingin mengelola data pengendara yang akan melakukan parkir di <i>restaurant</i> .
Pre-condition	:	1. Petugas parkir berada dalam jam kerja. 2. Petugas parkir ingin melakukan <i>log-in</i> kedalam sistem.
Post-condition	:	1. Petugas parkir mengisi <i>username</i> dan <i>password</i> .
Success scenario	:	1. Proses <i>log-in</i> berhasil.
Extension scenari	:	1. Petugas parkir lupa akan <i>username</i> ataupun <i>passwordnya</i> . 1a. Petugas parkir melakukan perubahan terhadap useraname atau <i>passwordnya</i> . 2. Petugas parkir salah mengisi <i>username</i> atau <i>password</i> . 2a. Pertugas parkir mengulang kembali pengisian <i>username</i> ataupun <i>password</i> .

Tabel 2. [UCS-02]-Melihat History Pengendara

Identifier	:	UC02
Goal	:	Sistem memunculkan informasi jumlah pengendara yang sudah parkir pada waktu tertentu.
Primary actor	:	Petugas Parkir.
Secondary actor	:	-
Trigger	:	Petugas parkir ingin melihat informasi mengenai jumlah pengendara yang telah parkir.
Pre-condition	:	Petugas parkir sudah login ke sistem.

Post-condition	:	Sistem memunculkan informasi mengenai pengendara yang telah parkir .
Success scenario	:	<ol style="list-style-type: none"> 1. Petugas parkir login ke sistem. 2. Petugs parkir membuka <i>History</i> Parkir Pelanggan. 3. Sistem memunculkan <i>history</i> mengenai informasi pengendara yang telah melakukan parkir.
Extension scenari	p:	<ol style="list-style-type: none"> 1. Sistem gagal memunculkan <i>database</i> mengenai informasi pelanggan. <ol style="list-style-type: none"> 1a. Petugas Parkir memeriksa kondisi jaringan 1b. Petugas parkir mengakses kembali <i>History</i> Parkir pengendara.

Tabel 3. [UCS-03]-Mendata Pengendara yang Parkir

Identifier	:	UC03
Goal	:	Pengendara berhasil memakir kendaraan di parkir DJ Ongkok <i>Restaurant</i> .
Primary actor	:	Pengendara .
Secondary actor	:	Petugas parkir.
Trigger	:	Seorang pengendara yang ingin memarkirkan kendaraannya dan terdapat parkir kosong.
Pre-condition	:	<ol style="list-style-type: none"> 1. Pengendara sedang mengakses sistem. 2. Pengendara berada di kawasan parkir DJ Ongkok <i>Restaurant</i>. 3. Tersedia slot parkir yang kosong.
Post-condition	:	<ol style="list-style-type: none"> 1. Pengendara memarkirkan kendaraannya di lokasi parkir yang kosong. 2. Timer untuk menghitung waktu parkir aktif setelah slot parkir berubah menjadi merah. 3. Pengendara menerima <i>kode Verifikasi</i> yang digunakan sebagai verifikasi

Success scenario	:	<ol style="list-style-type: none"> 1. Pengendara datang ke <i>restaurant</i>. 2. Pengendara melihat ada slot parkir berwarna hijau pada sistem. 3. Pengendara mengklik slot parkir berwarna hijau sehingga muncul form untuk informasi pengendara yang akan memarkir di slot tersebut. 4. Pengendara mengisi form dan menyubmit informasi pada form. 5. Warna slot parkir berubah menjadi merah. 6. Timer diaktifkan dan waktunya ditampilkan di atas slot.
		<ol style="list-style-type: none"> 7. Nomor plat ditampilkan di atas slot. 8. Sistem memberikan <i>kode Verifikasi</i> kepada pengendara yang digunakna sebagai verifikasi
Extension scenario:		<ol style="list-style-type: none"> 1. Informasi pengendara gagal disimpan ke database. <ol style="list-style-type: none"> 1a. Pengendara memeriksa kondisi jaringan. 1b. Pengendara kembali melakukan pengisian informasi

Tabel 4. [UCS-04]-Melihat Parkiran Kosong

Identifier	:	UC04
Goal	:	Menunjukan slot parkir yang kosong(hijau) dan tidak kosong(merah)
Primary actor	:	Pengendara.
Secondary actor	:	-
Trigger	:	Seorang pengendara yang ingin memarkirkan kendaraanya
Pre-condition	:	<ol style="list-style-type: none"> 1. Pengendara memasuki wilayah <i>restaurant</i>. 2. Pengendara melihat disistem dimana parkir yang kosong, dimana warna hijau menandakan kosong dan merah menandakan tidak kosong(berisi).

Post-condition	:	<ol style="list-style-type: none"> 1. Pengendara melihat lokasi parkir berwarna hijau (kosong) disistem. 2. Pengendara memarkirkan kendaraanya di lokasi parkir yang kosong.
Success scenario	:	<ol style="list-style-type: none"> 1. Pengendara datang ke <i>restaurant</i>. 2. Pengendara melihat dari sistem dimana lokasi parkir yang kosong (warna hijau). 3. Pengendara memarkirkan kendaraanya di lokasi parkir yang kosong.
Extension scenari	:	<ol style="list-style-type: none"> 1. Pengendara melihat disistem lokasi parkir penuh (warna merah). <ol style="list-style-type: none"> 1a. Pengendara membatalkan parkir di <i>restaurant</i>.

Tabel 5. [UCS-05]-Mendata Pengendara yang Keluar Parkiran

Identifier	:	UC05
Goal	:	Pengendara meninggalkan area parkir DJOR .
Primary actor	:	Pengendara.
Secondary actor	:	Petugas Parkir.
Trigger	:	Pengendara memarkirkan kendaraannya pada area parkir DJOR.
Pre-condition	:	<ol style="list-style-type: none"> 1. Pengendara memasuki area parkir DJOR apabila tersedia slot aria parkir yang kosong. 2. Pengendara telah selesai menikmati hidangan di DJOR.
Post-condition	:	<ol style="list-style-type: none"> 1. Pengendara telah meninggalkan area parkir DJOR. 2. Data pengendara telah dicatat kedalam daftar pemarkir DJOR.
Success scenario	:	<ol style="list-style-type: none"> 1. Pengendara memberikan <i>kode Verifikasi</i> (yang diterima saat masuk area parkir) kepada petugas parkir. 2. Petugas parkir melakukan verifikasi data pengendara, melalui <i>kode Verifikasi</i> dan plat kendaraan.

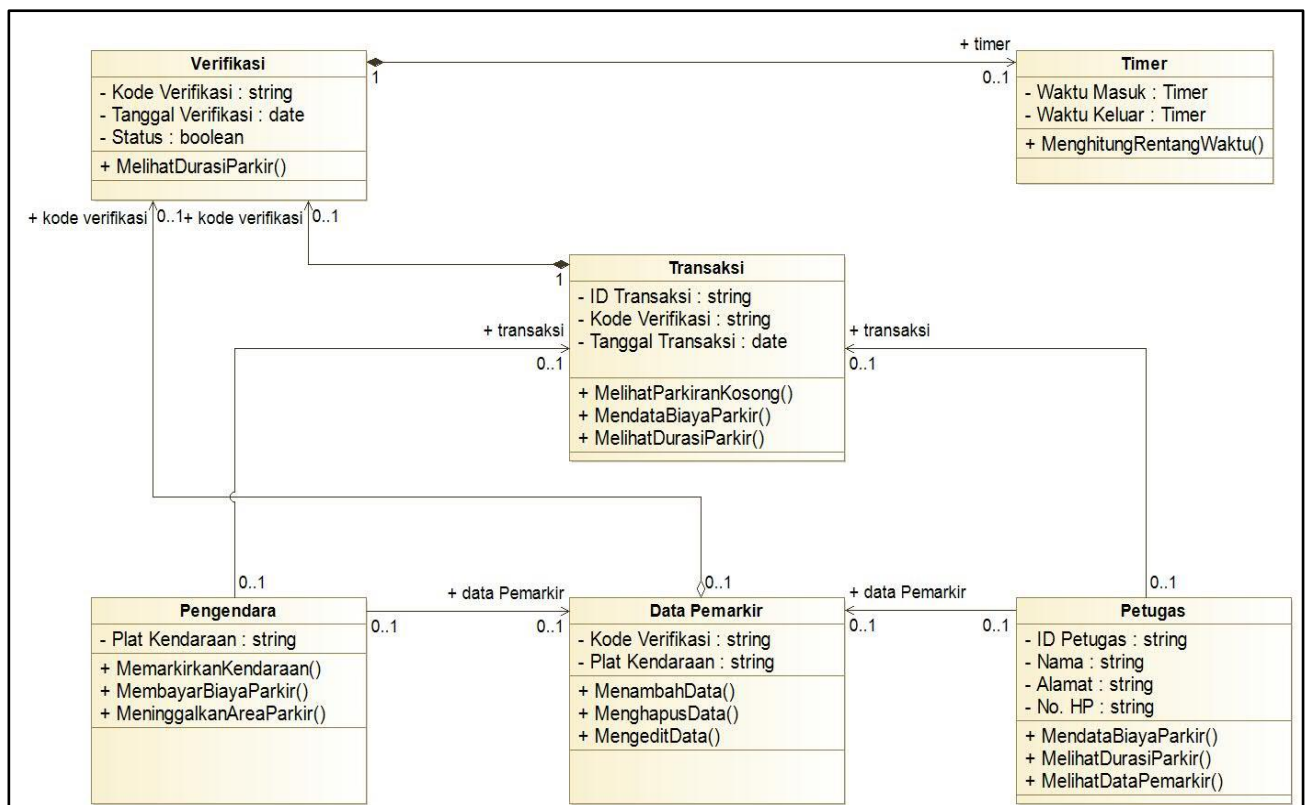
		3. Pengendara dapat meninggalkan area parkir DJOR.
Extention scenario	:	1. Pengendara kehilangan <i>kode Verifikasi</i> yang diterima saat memasuki area parkir. 2. Terdapat perbedaan data pengendara atau <i>kode Verifikasi</i> tertukar.

Tabel 6. [UCS-06]-Mendata Biaya Parkir

Identifier	:	UC06
Goal	:	Petugas Parkir dapat mengupdate daftar transaksi yaitu data pembayaran biaya parkir.
Primary actor	:	Petugas Parkir.
Secondary actor	:	Pengendara.
Trigger	:	Petugas parkir menerima biaya parkir pengendara.
Pre-condition	:	1. Terdapat pengendara yang memberikan biaya parkir sesuai biaya yang ditentukan. 2. Terdapat Petugas Parkir yang bertugas.
Post-condition	:	1. Petugas parkir menerima setoran biaya parkir Pendendara sesuai dengan tagihan biaya yang ditetapkan 2. Petugas melakukan pemasukan data pembayaran pada sistem 3. Data pembayaran tercatat pada sistem
Success scenario	:	1. Pengendara membayar tagihan biaya parkir sesuai <i>kode Verifikasi</i> 2. Petugas Parkir melakukan validasi data pengendara. 3. Pengendara memberikan uang bayar parkir sejumlah nominal yang di tetapkan . 4. Petugas Parkir menerima uang pembayaran dan mencatat log transaksi . 5. Petugas parkir melakukan input data transaksi pembayran yang berhasil.

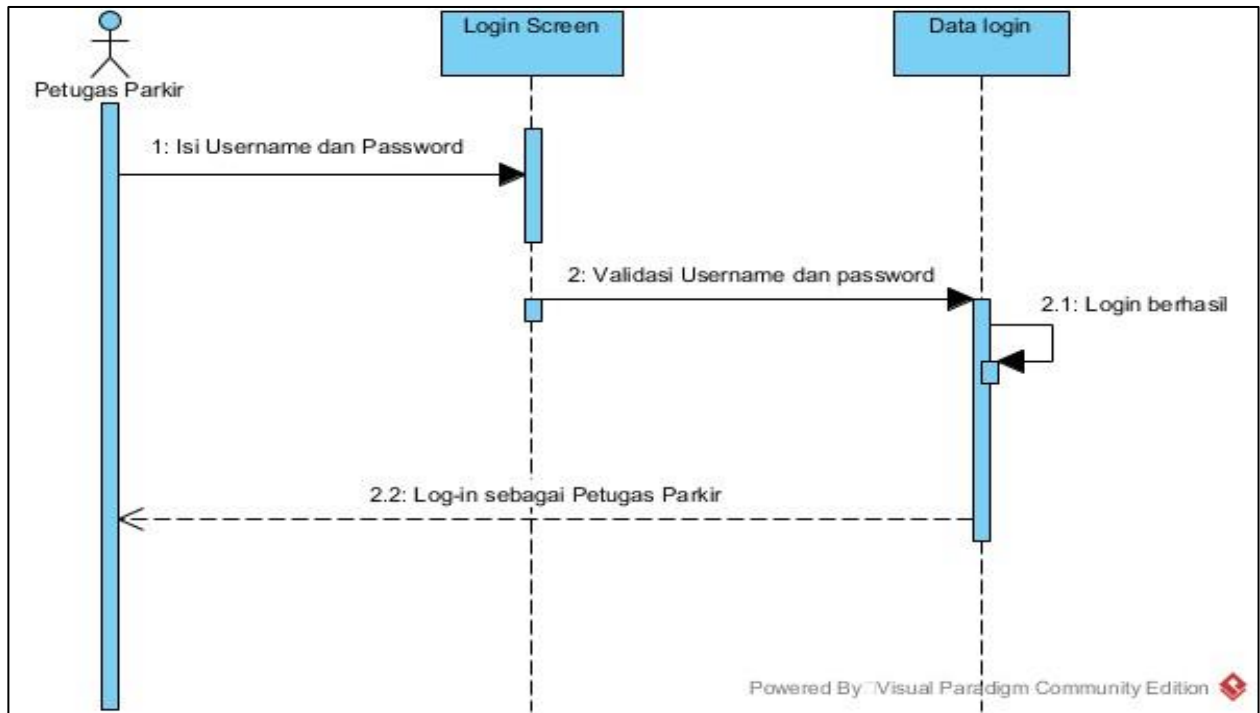
		6. System memberikan catatan laporan pembayaran.
Extension scenario	:	1. Pengendara membatalkan proses akses parkiran 1.a. Transaksi di batalkan . 1.b. Tidak ada parkiran kosong yang tersedia.

4. Class Diagram

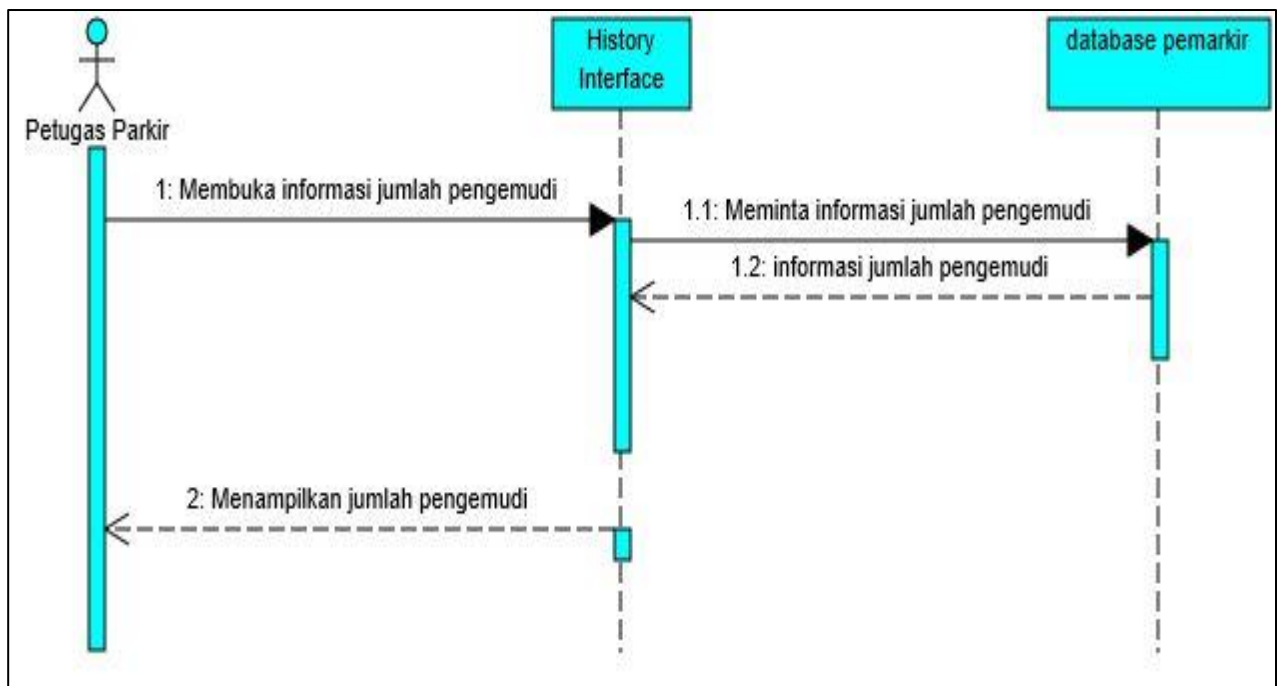


5. Sequence Diagram

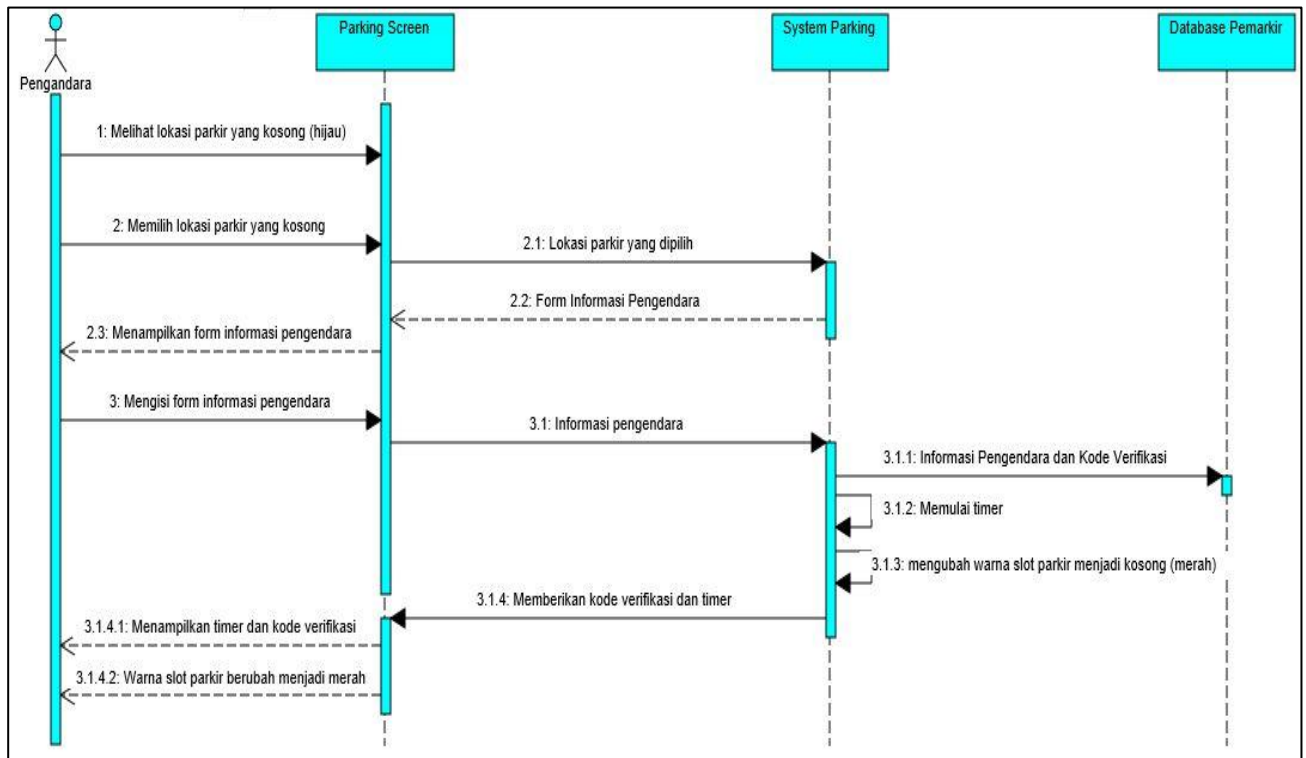
Sequence Diagram terhadap use case 01: 'Login'



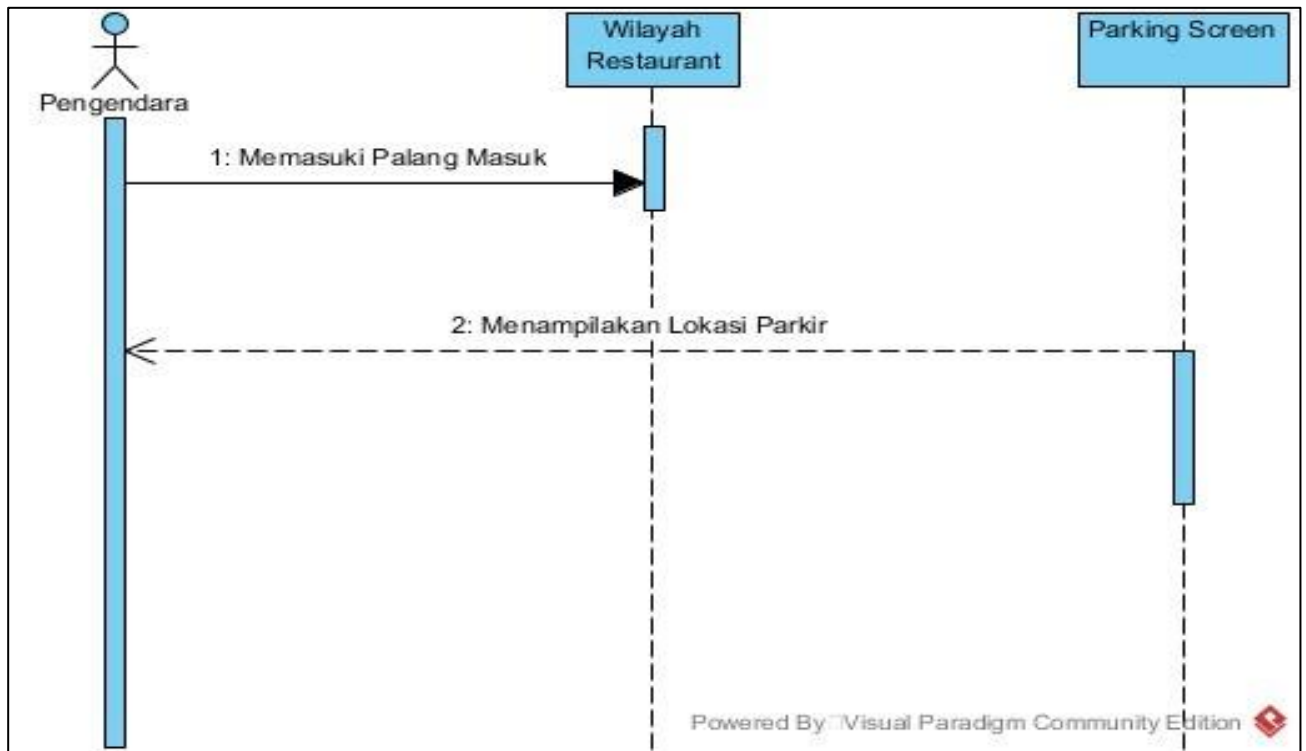
Sequence Diagram terhadap use case 02: 'Melihat History Pengendara'



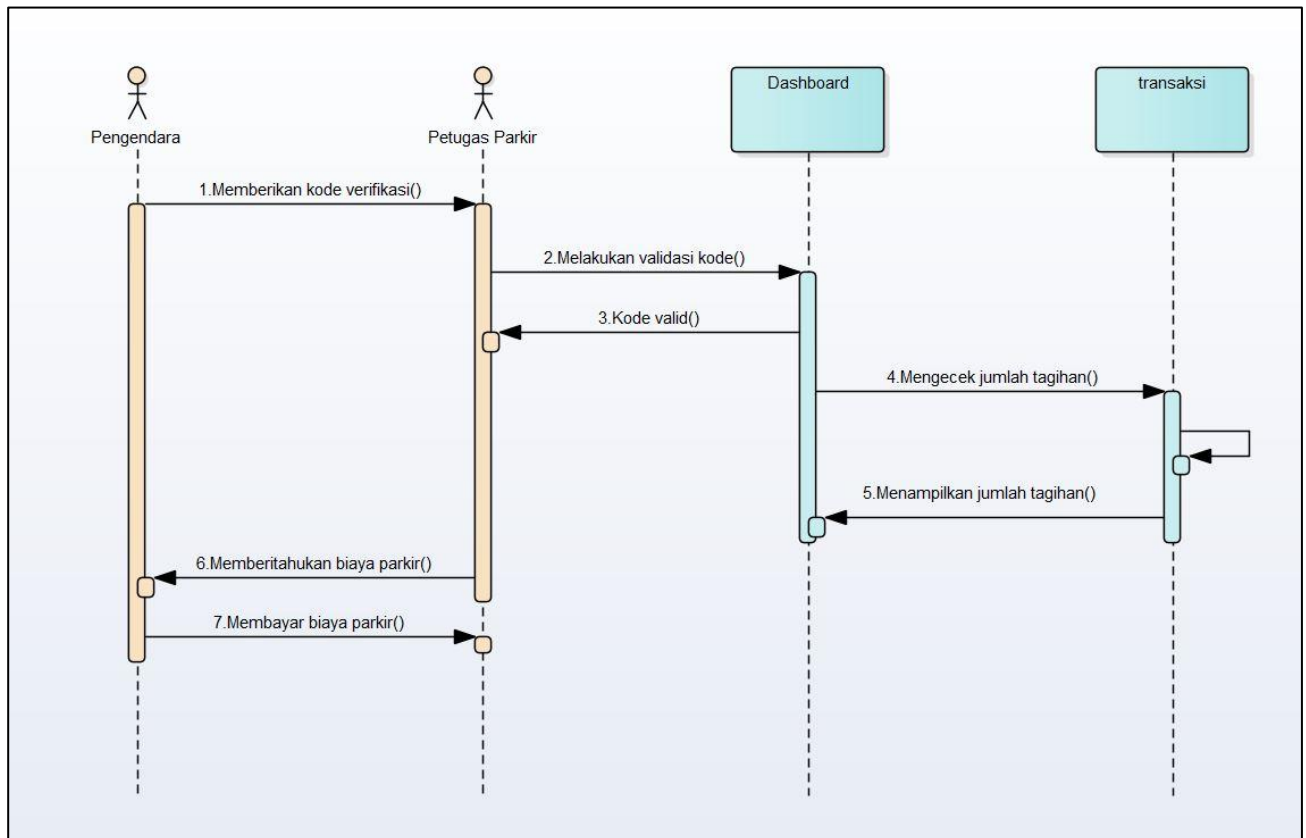
Sequance Diagram terhadap use case 03 : 'Mendata Pengendara yang Parkir'



Sequance Diagram terhadap use case 04 : 'Melihat Parkiran Kosong'



Sequance Diagram terhadap use case 05 : 'Mendata Pengendara yang Keluar Parkiran'



Sequance Diagram terhadap use case 06 : 'Mendata Biaya Parkir'

