

IMPLEMENTASI RESTFUL WEB SERVICES DENGAN OTORISASI OAUTH 2.0 PADA SISTEM PEMBAYARAN PARKIR

Indra Kusuma

Fakultas Ilmu Komputer, Program Studi Teknik Informatika
Universitas Dian Nuswantoro Semarang
Email: id.indrakusuma@gmail.com

Ajib Susanto

Fakultas Ilmu Komputer, Program Studi Teknik Informatika
Universitas Dian Nuswantoro Semarang
Email: ajib.susanto@dsn.dinus.ac.id

Ibnu Utomo Wahyu Mulyono

Fakultas Ilmu Komputer, Program Studi Teknik Informatika
Universitas Dian Nuswantoro Semarang
Email: ibnu.utomo.wm@dsn.dinus.ac.id

ABSTRAK

Salah satu jenis retribusi daerah adalah retribusi parkir. Setiap pemerintah daerah memiliki wewenang dalam menetapkan tarif retribusi parkir. Tetapi, sering sekali para juru parkir meminta lebih dari tarif yang sudah ditetapkan. Hal tersebut merugikan pengguna parkir, selain harus membayar lebih, pengguna parkir sering tidak mendapatkan karcis parkir. Belum lagi pengelolaan retribusi parkir yang tidak transparan menjadi salah satu penyebab rendahnya penerimaan retribusi parkir. Permasalahan ini dapat diminimalkan dengan menerapkan teknologi RESTful web services pada sistem pembayaran retribusi parkir yang terintegrasi dan transparan sehingga diharapkan mampu menjadi alternatif solusi mengatasi masalah pengelolaan retribusi parkir yang dialami oleh Pemkot maupun Pemda di Indonesia. Metode pengembangan sistem yang digunakan yaitu Extreme Programming (XP) dengan tahap : planning (perencanaan), design (perancangan), coding (pengkodean) dan testing (pengujian). Hasil aplikasi berbasis Android yaitu Markir sebagai sarana utama mengirimkan data pembayaran parkir dan OAuth 2.0 sebagai protokol otorisasi setiap transaksi data yang dilakukan. Hasil pengujian aplikasi Markir dengan metode black-box testing, sistem telah terintegrasi dengan baik dan dilakukan berbagai tahap validasi seperti client credentials, access token dan scope akses fungsi yang tersedia pada API Markir sesuai standar protokol OAuth 2.0 sehingga transaksi data lebih aman jika dibandingkan menggunakan otorisasi API standar seperti basic-auth API ataupun API Key.

Kata kunci: retribusi parkir; android; RESTful *web services*; OAuth 2.0.

ABSTRACT

One type of regional retribution is parking fees. Every local government has the authority to set parking fees. However, very often parking attendants request more than the set rate. This is detrimental to parking users, besides having to pay more, parking users often do not get parking tickets. Not to mention the management of parking fees that are not transparent is one of the causes of the low acceptance of parking fees. This problem can be minimized by implementing RESTful web services technology in an integrated and transparent parking retribution payment system so that it is expected to be an alternative solution to overcome parking retribution management problems experienced by the City Government and Local Government in Indonesia. System development methods used are Extreme Programming (XP) which includes planning (planning), design (design), coding (coding) and testing (testing). The results of the Android-based application, Marker, are the main means of sending parking payment data and OAuth 2.0 as the protocol for authorizing each data transaction. The results of the Markir application testing with the black-box testing method, the system has been well integrated and carried out various stages of validation such as client credentials, access tokens and scope of access functions available on the API API according to the OAuth 2.0 protocol standard so that data transactions are safer than using authorization Standard APIs such as basic-auth API or API Key

Keywords: parking retribution; android; RESTful *web services*; OAuth 2.0.

1. PENDAHULUAN

Pendapatan Asli Daerah (PAD) memiliki peran penting dalam pembiayaan kegiatan pemerintahan dan pembangunan daerah. Sumber pendapatan asli daerah antara lain adalah pajak daerah dan retribusi daerah dimana setiap daerah diberikan kewenangan untuk melaksanakan pemungutan berbagai jenis pajak daerah dan retribusi daerah yang berkaitan dengan berbagai aspek kehidupan masyarakat[1]. Salah satu jenis retribusi daerah adalah retribusi parkir. Setiap pemerintah daerah memiliki wewenang dalam menetapkan tarif retribusi parkir. Pada saat penelitian ini dibuat, Pemkot Semarang menerapkan tarif parkir untuk kendaraan bermotor roda dua sebesar Rp. 1.000 (Seribu Rupiah), kendaraan bermotor roda tiga sebesar Rp. 1.500 (Seribu lima ratus rupiah), kendaraan bermotor roda empat sebesar Rp. 2.000 (Dua ribu rupiah), kendaraan bermotor roda enam sebesar Rp. 4.000 (Empat ribu rupiah) dan kendaraan bermotor roda lebih dari enam sebesar Rp. 7.000 (Dua ribu rupiah)[2].

Tetapi dalam praktiknya, banyak dari juru parkir meminta pengguna parkir untuk membayar retribusi parkir melebihi tarif retribusi yang berlaku dan tidak memberikan karcis parkir kepada pengguna parkir[3]. Beragam cara dilakukan oleh juru parkir dalam mendapatkan tarif retribusi parkir melebihi peraturan perda yang berlaku. Salah satunya adalah dengan cara yang sering dilakukan yaitu dengan tidak memberikan uang kembalian kepada pengguna parkir[4]. Tentu saja hal tersebut merugikan pengguna parkir. Karena selain harus membayar lebih, pengguna parkir juga sering tidak mendapatkan karcis parkir yang sudah menjadi haknya.

Pada tahun 2016 target PAD sektor retribusi parkir Pemkot Semarang sebesar 4 miliar tidak tercapai[5]. Hal tersebut dikarenakan Pemkot Semarang dinilai belum maksimal sehingga terdapat kebocoran dalam pengelolaan retribusi parkir. Sehingga dalam hal ini, tidak hanya pengguna parkir saja yang dirugikan, tetapi pemerintah kota Semarang juga dirugikan. Berdasarkan penelitian yang telah dilakukan oleh Betti Arvita pada tahun 2015, salah satu penyebab rendahnya realisasi penerimaan retribusi parkir oleh pemerintah kota adalah pengelolaan parkir yang tidak transparan[6]. Hal tersebut ditandai dengan tidak relevannya antara setoran penerimaan retribusi parkir yang masuk ke pemerintah kota dengan karcis yang dibagikan atau yang terjual.

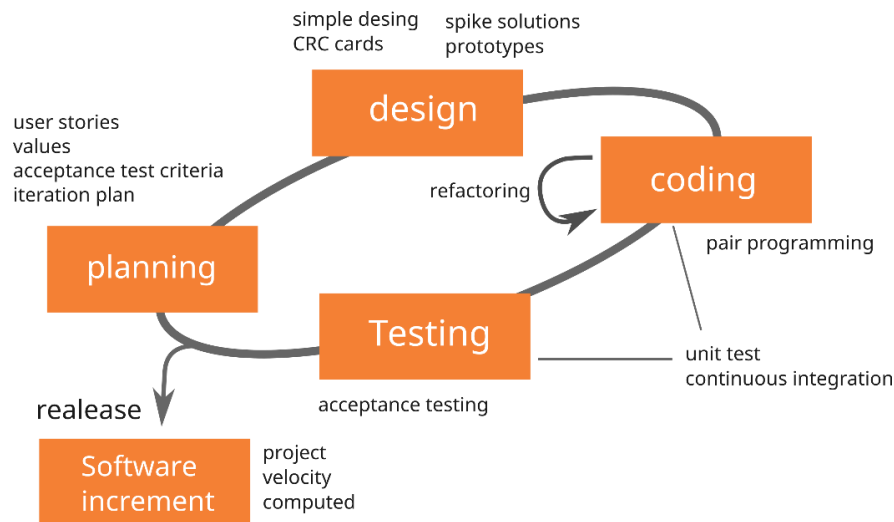
Salah satu teknologi yang dapat diterapkan dalam mengatasi berbagai masalah tersebut adalah teknologi *Web Services*. Dengan menggunakan teknologi *Web Services*, seperti yang digunakan untuk proses perhitungan suara secara cepat pada Pilkada[7] dan *Web Services* untuk menghubungkan sistem server pulsa dengan *Yahoo Messenger*(YM)[8] maka antar satu sistem dengan sistem lainya dapat berkomunikasi walaupun berbeda platform maupun bahasa pemrograman. Dalam berkomunikasi melalui *Web Services*, diperlukan protokol otorisasi pada sistem agar komunikasi data terjamin keamanannya. Salah satu protokol otorisasi yang saat ini direkomendasikan dan digunakan oleh perusahaan besar seperti *Facebook*, *Google* dan *Microsoft* adalah *OAuth*[9].

Berdasarkan permasalahan dan fakta yang telah penulis jelaskan sebelumnya, maka dibutuhkan sebuah sistem yang dapat memberikan layanan dengan kemudahan dan keamanan yang terjamin yaitu aplikasi berbasis web dan *Android* dengan mengimplementasikan *RESTful Web Services* dengan Protokol Otorisasi *OAuth 2.0* pada Sistem Pembayaran Parkir yang terintegrasi dan transparan. Sehingga diharapkan mampu menjadi alternatif solusi untuk mengatasi permasalahan pada pengelolaan retribusi parkir yang dialami oleh pemerintah kota di Indonesia khususnya kota Semarang.

2. METODOLOGI PENELITIAN

2.1 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan yaitu *Extreme Programming (XP)*[10][11] dengan pemodelan *Unified Modeling Language (UML)* yang menitikberatkan pada interaksi antara *customer* dengan *developer*. Aplikasi yang dikerjakan sesuai dengan kesepakatan antar *customer* dan *developer* dalam suatu iterasi dimana tiap iterasinya menghasilkan suatu unit aplikasi yang siap digunakan (ditingkatkan fungsinya untuk iterasi berikutnya, dan terus diperbaiki hingga tujuan penelitian yang diinginkan sepenuhnya tercapai)[12].



Gambar 1. Langkah Extreme Programming (XP)[10]

Langkah pengembangan sistem menggunakan XP adalah :

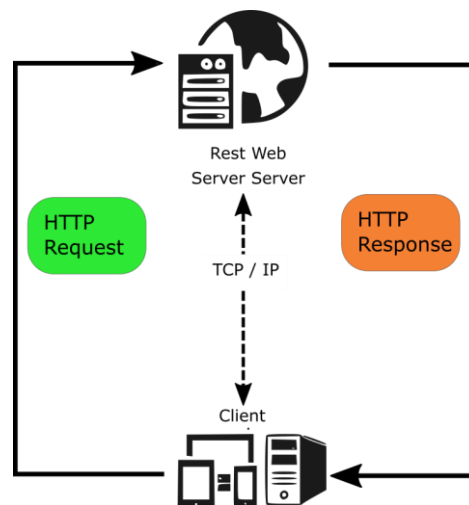
- Release Planning* (perencanaan penelitian secara utuh)
- Iteration* (iterasi penelitian, terdapat perencanaan, pengerjaan, serta pengujian di tiap unit aplikasi yang dihasilkan)
- Acceptance Test* (pengujian secara menyeluruh untuk versi terakhir aplikasi)
- Small Release* (rilis aplikasi)

2.2 Web Services

Web Services adalah suatu teknologi yang dirancang sebagai upaya mengatasi interoperabilitas antar satu sistem dengan sistem lainnya dalam suatu jaringan. *Web Services* merupakan suatu fitur yang disediakan oleh penyedia layanan berupa layanan informasi untuk sistem yang lain sehingga antar sistem dapat saling berinteraksi melalui *services* yang diberikan[13]. *Web Services* menyimpan data informasi dalam format standar seperti *HTTP*, *SSL*, *SOAP*, *XML*, *JSON*[7].

2.3 RESTful Web Services

REST (Representational State Transfer) adalah salah jenis *web services* populer. *REST* juga sering disebut sebagai *RESTful*. *REST* dapat digambarkan seperti saat mengakses alamat web, *browser* akan meminta halaman web dan kemudian server web tersebut akan mengirimkan *State* halaman web ke *browser*. Metode ini berorientasi pada sumber daya informasi (*resource*) dan memiliki keluaran berupa *XML* atau *JSON*[14].



Gambar 2. Interaksi RESTful Web Services

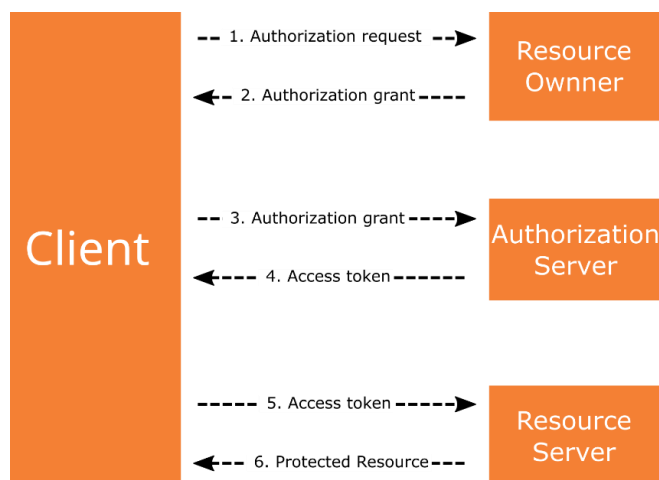
Cara interaksi *RESTful* yang digambarkan pada Gambar 2 yang dilakukan oleh *client* dan direspon oleh *server* dengan format *JSON*. Berikut merupakan format respon *JSON* dari server.

```
{
  "success": true,
  "message": "Data ditemukan",
  "data": {
    "nama_kendaraan": "Beat Merahku",
    "plat_nomor": "H-1123-VM",
    "tipe": "BeAt",
    "vendor": "Honda",
    "roda": 2
  }
}
```

Gambar 3. Contoh respon dengan format JSON

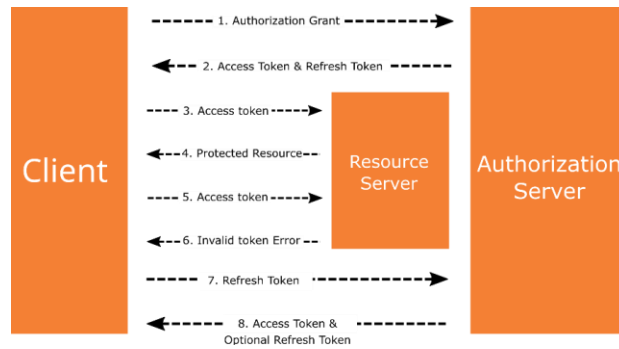
2.4 OAuth 2.0

OAuth merupakan suatu protokol terbuka yang mengizinkan otorisasi secara aman dengan metode yang sederhana dan standar dari aplikasi web, *mobile* dan desktop[15]. Protokol otorisasi *OAuth* memungkinkan aplikasi pihak ketiga (*third-party*) untuk dapat mengakses sumber daya tertentu seperti foto, video maupun informasi tertentu dan dibatasi dengan akses tertentu melalui protokol *HTTP*. *OAuth* dirancang khusus untuk mengatasi permasalahan klasik pada model autentikasi pada *client-server*[16]. Gambar 4 menunjukkan proses otorisasi yang dilakukan menggunakan protokol *OAuth 2.0*.



Gambar 4. OAuth2.0 Flow Protocol

Untuk menjaga keamanan kredensial, *access token* yang diberikan kepada client memiliki masa kadaluwarsa. Ketika masa berlaku sudah habis maka diperlukan untuk meminta *access token* yang baru kepada *authorization provider* atau *OAuth provider*. *Refresh Token* merupakan kredensial yang digunakan untuk mendapatkan *access token* yang sudah kadaluwarsa. Gambar 5 berikut merupakan *flow of Refresh Token*[17].



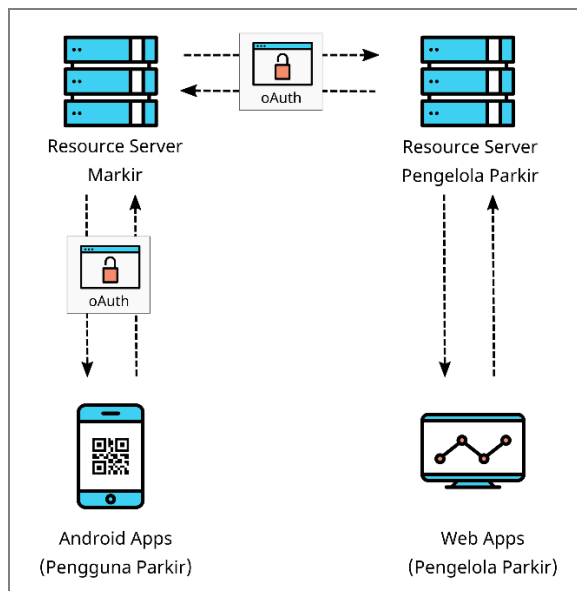
Gambar 5. Flow of Refresh Token

3. HASIL DAN PEMBAHASAN

3.1 Sistem Pembayaran Parkir

Retribusi parkir merupakan pungutan daerah yang merupakan salah satu bentuk pembayaran atas jasa atau pemberian izin memberhentikan kendaraan pada suatu tempat parkir di tepi jalan umum dalam jangka waktu tertentu yang disediakan dan/atau diberikan oleh Pemerintah Daerah untuk kepentingan orang pribadi atau badan[2]. Pembayaran parkir saat ini masih menjadi masalah karena banyak dari petugas parkir meminta tarif parkir melebihi tarif yang diatur oleh peraturan daerah[3]. Markir merupakan nama dari sistem pembayaran parkir berbasis Android. Markir dibangun sebagai alternatif solusi untuk permasalahan tentang transparansi pembayaran retribusi parkir yang dikelola oleh Pemerintah Daerah.

Sistem Markir ini terdiri dari sisi server (*server side*) dan sisi klien (*client side*) berupa satu aplikasi Android serta dua aplikasi web. Interaksi antar bagian menggunakan teknologi RESTful web service melalui protokol HTTP. Bagian klien akan melakukan HTTP Request ke server, dan server akan melakukan otorisasi dengan protokol OAuth 2.0, setelah otorisasi berhasil dilakukan maka server akan memberikan respon berupa data berformat JSON, data respon tersebut kemudian akan digunakan oleh bagian klien sesuai dengan kebutuhan. Berikut merupakan gambar dari alur dari sistem Markir.



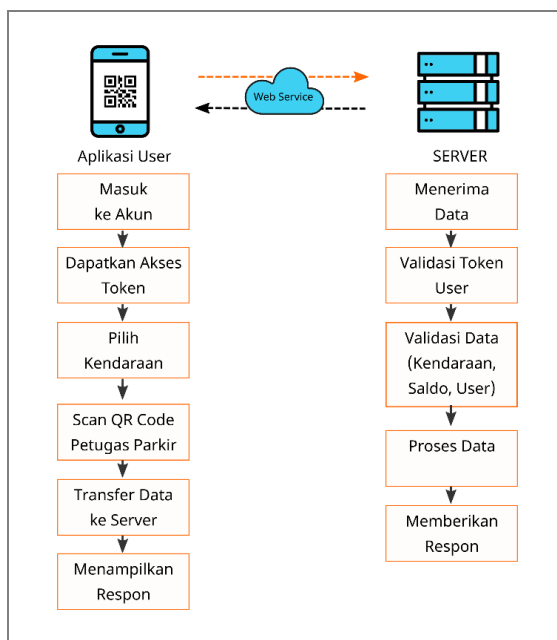
Gambar 6. Alur Sistem Markir

Berikut merupakan penjelasan dari Alur dari Sistem Pembayaran Retribusi Parkir yang digambarkan pada Gambar 6.

- Android Apps merupakan aplikasi yang dikembangkan pada bagian klien (*client-side*) untuk pengguna parkir. Aplikasi ini mengirim data kredensial pengguna parkir pada OAuth untuk melakukan autentikasi dan otorisasi melalui *REST web service* dengan format JSON.

- b) OAuth akan memeriksa kredensial yang dikirimkan pengguna parkir, jika sesuai maka OAuth akan memberikan otorisasi melalui access token (kode token) melalui *REST web services*.
- c) Jika pengguna parkir sudah mendapatkan *access token* dari OAuth melalui *REST web services*, pengguna parkir bisa menggunakan aplikasi sesuai fitur utamanya.
- d) Pengguna parkir dapat melakukan pembayaran retribusi parkir dengan mengirimkan dan menerima data melalui *REST web services* pada *Resource Server Markir*.
- e) Web Apps atau pengelola parkir adalah pemilik lahan parkir yang bisa memantau aktivitas dan pendapatan parkir melalui Web Apps pada server pengelola parkir yang sudah terotorisasi dengan *Resource Server Markir*.

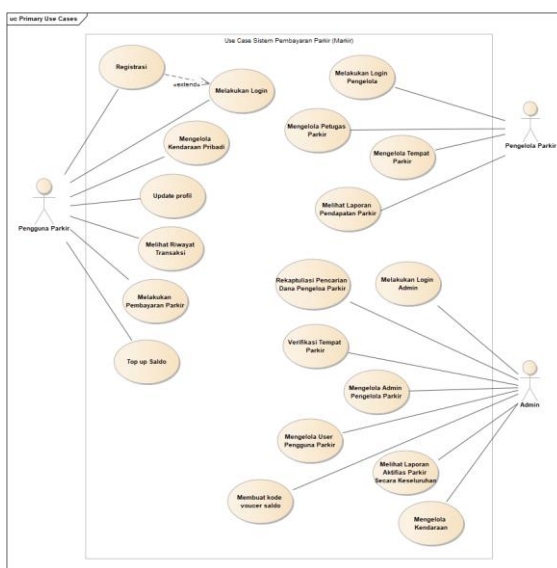
Proses utama yang menjadi bagian inti dari adalah proses pembayaran retribusi parkir. Gambar 7 berikut ini merupakan gambaran alur dari proses pembayaran retribusi parkir.



Gambar 7. Alur Dari Proses Pembayaran Retribusi Parkir Pada Sistem Markir

3.2 Perancangan Sistem

- a) Diagram *Use Case*



Gambar 8. Diagram Use Case Markir

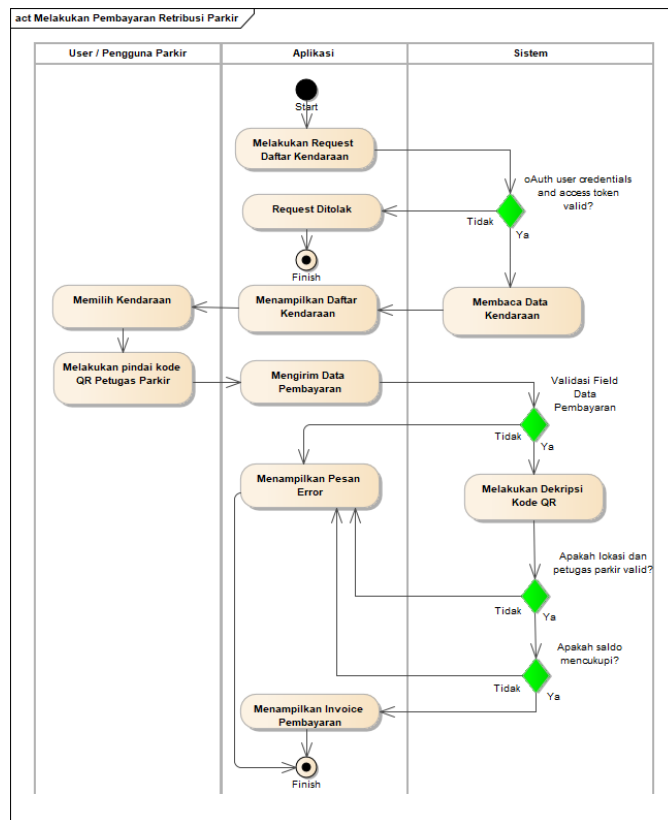
Dari *use case* yang digambarkan pada Gambar 8 menunjukan bahwa aktor yang terlibat pada sistem Markir ini terdiri dari tiga aktor utama yaitu pengguna parkir, pengelola parkir, serta admin. Aktor Pengguna parkir merupakan aktor pada sisi klien dengan aplikasi berbasis Android minimal versi 4.0.4 dari sistem pembayaran retribusi parkir. Dengan menggunakan aplikasi ini, maka pengguna bisa melakukan registrasi akun, mengelola informasi, melakukan *top up* saldo dan melakukan pembayaran retribusi parkir kepada mitra yang sudah bekerja sama.

Sedangkan Pengelola Parkir merupakan aktor pada sisi klien dengan aplikasi berbentuk *Web App* (Web Aplikasi) dari sistem pembayaran retribusi parkir. Dengan melakukan otorisasi *username* dan *password* yang dari setiap akun Pengelola Parkir. Maka pengelola dapat menggunakan aplikasi ini untuk melihat laporan pendapatan parkir, mengelola petugas parkir, mengelola lahan parkir dan mengajukan pencairan dana.

Aktor yang terakhir adalah Admin. Bertugas untuk mengelola data yang digunakan oleh sistem pembayaran retribusi parkir seperti mengelola data kendaraan (merek dan jenis), verifikasi tempat parkir, verifikasi pencairan dana, mengelola Pengelola Parkir, mengelola Pengguna Parkir.

b) Activity Diagram

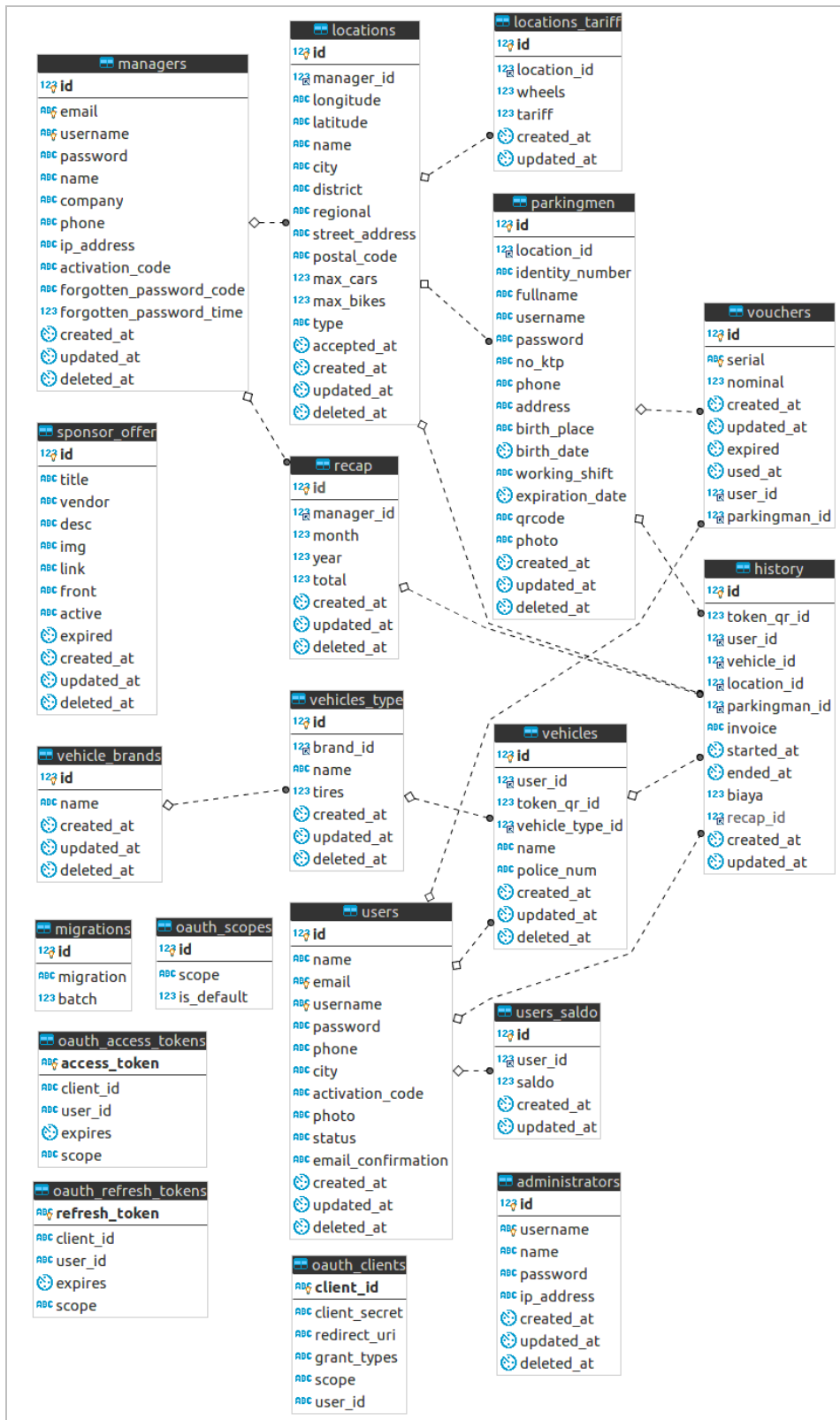
Proses inti dari sistem Markir adalah pembayaran retribusi parkir. Berikut merupakan *activity diagram* dari proses pembayaran parkir.



Gambar 9. Activity Diagram Pembayaran Retribusi Parkir

c) Desain Database

Hubungan antar tabel dalam database Sistem Pembayaran Retribusi Parkir yang digunakan pada sistem Markir berbasis android dengan menggunakan *Rest Web Services* dan *OAuth 2.0* terlihat pada gambar 10 berikut ini :



Gambar 10. Relasi Tabel Pada Sistem Pembayaran Retribusi Parkir

3.3 Implementasi Sistem

Setelah melakukan analisa serta perancangan sistem, langkah selanjutnya adalah implemementasi pada program yang akan dibuat.

a) OAuth 2.0

```
public function handle($request, Closure $next)
{
    # OAuth check for user scope..
    if ($request->is('api/users/*')) {
        1 $this->oauth2 = new OAuth([
            'storage' => ['user_table' => 'users']
        ]);
        $this->oauth2->require_scope('userinfo user');
    }
    # OAuth check for officer check..
    elseif ($request->is('api/officers/*')) {
        2 $this->oauth2 = new OAuth([
            'storage' => ['user_table' => 'parkingmen']
        ]);
        $this->oauth2->require_scope('userinfo officer');
    }
    # OAuth check for Developer mode..
    else {
        3 $this->oauth2 = new OAuth();
        $this->oauth2->require_scope('developer');
    }
    return $next($request); 4
}
```

Gambar 11. Implementasi OAuth 2.0 pada RESTful Services Markir

Gambar 11 merupakan pengkodean dari otorisasi setiap request yang diminta melalui RESTful Web Services dengan protokol otorisasi OAuth 2.0.

b) RESTful Web Services

```
public function payment(Request $request)
{
    # STEP 1
    1 $validator = Validator::make($request->all(), [
        'id_user' => 'required',
        'id_kendaraan' => 'required',
        'data' => 'required',
    ]);
    # STEP 2 | validasi
    2 if ($validator->fails()) {
        $this->respons = [
            4 'sukses' => false,
            'pesan' => 'Data yang dimasukkan belum lengkap.',
        ];
        return $this->respons;
    } 3
    $id_user = $request->input('id_user');
    $id_kendaraan = $request->input('id_kendaraan');
    $dataQR = $request->input('data');
    5 $this->qrcode = new QRcode();
    $this->invoice = new Invoice();
    $date = date("Y-m-d H:i:s");
    $qr_decrypted = $this->qrcode->decrypt_officer($dataQR);
    $data = explode('_', $qr_decrypted);
    $invoice = $this->invoice->generate($id_kendaraan, $data[2], $date);
    $location_type = Location::find($data[1])->value('type');
```

Gambar 12. Web Services Pembayaran Retribusi Parkir (PHP) – bagian 1

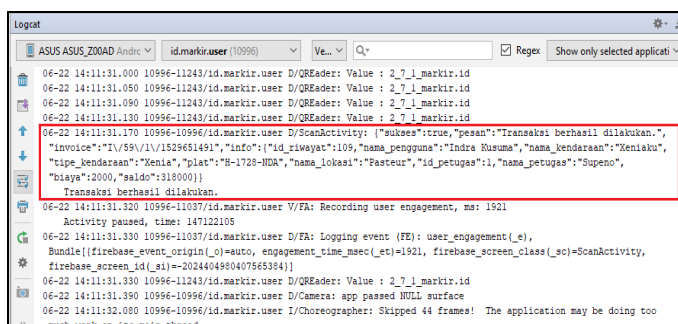
```
6 # check parking type?
  if ($location_type == 'F') {
    $biaya = MasterUser::parking_cost($data[1], $id_kendaraan)->first(); 10
    $user_saldo = MyUserSaldo::where('user_id', $id_user)->value('saldo'); 11
    # is saldo enough?
    12 if (($user_saldo - $biaya->biaya) < 0) {
      $this->respons = [
        14 'sukses' => false,
        'pesan' => 'Saldo anda tidak mencukupi. Harap isi ulang saldo terlebih dahulu.',
      ];
      return $this->respons;
    } 13
    # set data to history
    $data_post = [
      'user_id' => $id_user,
      'vehicle_id' => $id_kendaraan,
      'location_id' => $data[1],
      15 'parkingman_id' => $data[2],
      'invoice' => $invoice,
      'started_at' => $date,
      'biaya' => $biaya->biaya
    ];
    # is success to make history transaction?
    16 if (!History::create($data_post)) {
      $this->respons = [
        'sukses' => false,
        18 'pesan' => 'Data pembayaran belum tersimpan pada sistem. Harap coba kembali.',
      ];
      return $this->respons;
    } 17
    # is saldo updated?
    19 if (!MasterUser::parking_payment($id_user, $biaya->biaya)) {
      $this->respons = [
        'sukses' => false,
        21 'pesan' => 'Saldo pengguna belum terupdate.',
      ];
      return $this->respons;
    } 20
  }
```

Gambar 13. Web Services Pembayaran Retribusi Parkir (PHP) – bagian 2

```
    # get invoice data
    22 $data_info = MasterUser::get_after_scan($invoice)->first();
    $this->respons = [
      'sukses' => true,
      'pesan' => 'Transaksi berhasil dilakukan.',
      'invoice' => $invoice,
      'info' => $data_info,
    ];
    # end of transaction successfully 23
    return $this->respons;
  }
  # for now.. this type not supported!
  elseif ($location_type == 'NF') {
    $this->respons = [
      8 'sukses' => false,
      'pesan' => 'Saat ini parkir Non-Flat belum bisa dilakukan.',
    ];
    return $this->respons;
  }
  # parking type undefined
  else {
    $this->respons = [
      9 'sukses' => false,
      'pesan' => 'Lokasi tidak terdaftar.',
    ];
    return $this->respons;
  }
}
```

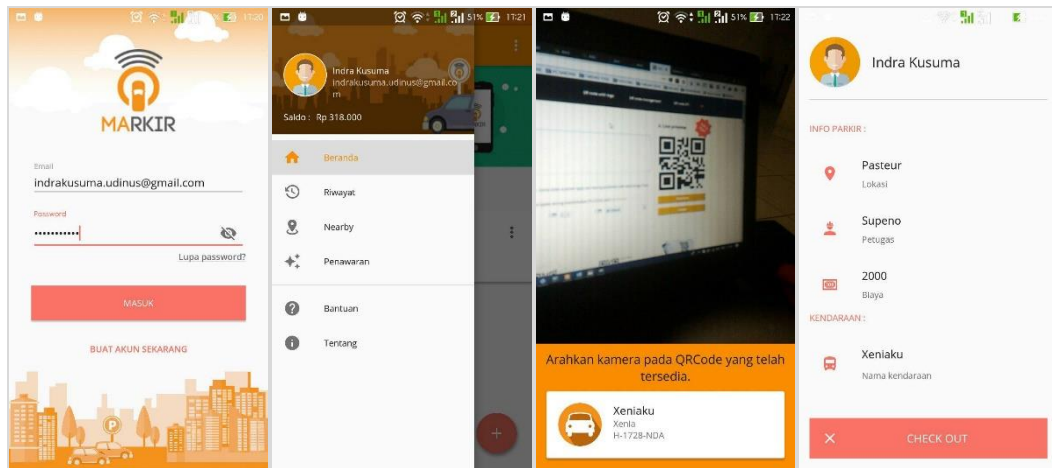
Gambar 14. Web Services Pembayaran Retribusi Parkir (PHP) – bagian 3

Gambar 12, Gambar 13, dan Gambar 14 menunjukkan pengkodean dari API pembayaran retribusi parkir yang bisa diakses melalui aplikasi klien Markir berbasis Android.



Gambar 15. Contoh Keterangan Gambar. (*inf_figurecaption*)

Gambar 15 menunjukkan respons dari *web services* yang diterima setelah proses pembayaran retribusi parkir pada aplikasi Markir.




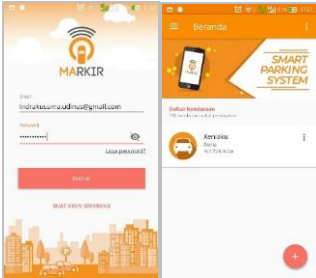

Gambar 16. Hasil Implementasi di Aplikasi Android



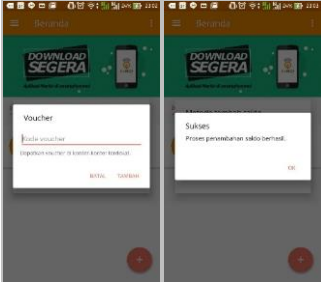

Gambar 16 menunjukkan hasil implementasi sistem pada aplikasi Markir pada sistem operasi Android. Dimulai dari kiri adalah halaman *log in*, kemudian *navigation*, *activity* pemindaahan QR petugas parkir serta yang paling kanan adalah *invoice* dari pembayaran parkir.

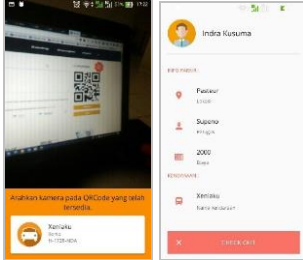
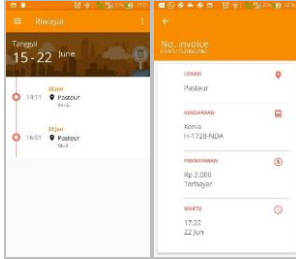

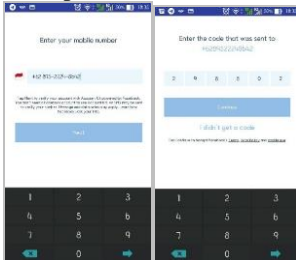
3.4 Pengujian Sistem

Berikut ini merupakan tabel hasil pengujian aplikasi untuk pengguna parkir menggunakan *Black-Box Testing* :

Tabel 1. Black-Box testing aplikasi untuk pengguna parkir

No.	Modul	Hasil yang diharapkan	Hasil Pengujian
1.	Splash Screen	Menampilkan Halaman splash screen saat membuka aplikasi.	OK
			
2.	Log in	Menampilkan halaman log in & Masuk ke halaman beranda	OK
			
3	Sidebar Menu	Dapat mengakses <i>Sidebar</i> menu, dan berpindah ke halaman <i>activity</i> lain.	OK
			

<i>No.</i>	<i>Modul</i>	<i>Hasil yang diharapkan</i>	<i>Hasil Pengujian</i>
4	Floating Button Menu	Dapat mengakses <i>floating button</i> menu serta dapat melakukan berpindah / membuka <i>event</i> lain.	OK
			
5	Menambah kendaraan	Menampilkan form penambahan kendaraan.	OK
			
		Serta dapat Menyimpan kendaraan dan menampilkan kendaraan yang baru disimpan.	
6	Melakukan Top Up Saldo	Menampilkan form top up saldo dan Jika kode voucher benar, Saldo ditambahkan ke akun pengguna.	OK
			
7	Profil	Dapat menampilkan halaman profil pengguna.	OK
			
		Serta dapat melakukan pembaharuan data.	

No.	Modul	Hasil yang diharapkan	Hasil Pengujian
8	Pembayaran Retribusi Parkir	Membuka halaman untuk melakukan pemindahan kode QR petugas parkir.	OK
			
		<ul style="list-style-type: none"> - Dapat mengenali kode QR petugas parkir yang. - Berhasil melakukan pembayaran parkir. Saldo berkurang. - Menampilkan halaman <i>invoice</i> dari pembayaran parkir. 	
9	Riwayat Transaksi	Menampilkan daftar riwayat transaksi parker & Menampilkan detail riwayat transaksi parkir.	OK
			
10	Log out	Dapat melakukan log out.	OK
11	Registrasi	Menampilkan halaman registrasi ketika di tekan tombol Buat Akun Sekarang.	OK
			
		Menampilkan form untuk input nomor telepon, serta konfirmasi nomor telepon.	
			

4. KESIMPULAN

Berdasarkan pembahasan sebelumnya dapat diambil kesimpulan bahwa sistem pembayaran retribusi parkir yang terintegrasi dan transparan dapat dibangun dengan konsep *RESTful web services* dengan otorisasi protokol *OAuth 2.0*, sehingga dapat memudahkan pengguna parkir dalam melakukan pembayaran retribusi parkir sesuai dengan tarif yang berlaku serta memudahkan petugas maupun pengelola parkir dalam memantau dan pencatatan transaksi retribusi parkir.

Pada penelitian ini telah dilakukan pengujian dengan metode *black-box testing* dan *white-box testing* terhadap sistem yang telah terintegrasi dengan baik serta dilakukan berbagai tahap validasi seperti *client credentials*, *access token* maupun *scope* untuk bisa mengakses fungsi yang tersedia pada API Markir sesuai dengan standar protokol *OAuth 2.0*. Sehingga bisa disimpulkan, bahwa penerapan *OAuth 2.0* lebih aman jika dibandingkan menggunakan otorisasi API standar seperti *basic-auth API* ataupun *API Key*.

DAFTAR PUSTAKA

- [1] Presiden Republik Indonesia, "UU Nomor 33 Tahun 2004 tentang Perimbangan Keuangan Antara Pemerintah Pusat dan Pemerintah Daerah." Negara Kesatuan Republik Indonesia, 2004.
- [2] Pemerintah Kota Semarang, "Peraturan Daerah Kota Semarang Nomor 2 Tahun 2012 Tentang Retribusi Jasa Umum Di Kota Semarang." Pemerintah Kota Semarang, 2012.
- [3] B. B. Budiasto, "Saya Disuruh Bayar Parkir Rp.5000 Padahal di Karcis Parkir Tertulis Rp 2.000," *Tribun Jateng*. .
- [4] "Tarif Parkir Tak Sesuai Perda," *Jawa Pos RADAR SEMARANG*. .
- [5] I. N. Layla, "10 Tahun, PAD Parkir Kota Semarang Tidak Maksimal," *Jawa Pos Radar Semarang*. .
- [6] A. Betti, "Analisis Faktor Faktor Penyebab Rendahnya Realisasi Penerimaan Retribusi Parkir Kota Padang," *Univ. Negeri Padang*, 2015.
- [7] F. N. Rofiq and A. Susanto, "Implementasi RESTful Web Service untuk Sistem Penghitungan Suara Secara Cepat pada Pilkada," *Eksplora Inform.*, pp. 159–168, 2017.
- [8] A. Ikrom, A.F.; Susanto, "Implementasi REST Web Service Pada Aplikasi Pengolah Pesan Yahoo Messenger (YM) Pada CV. Meliana Pratama," no. 1, pp. 27–36.
- [9] E. Chen, Y. Pei, S. Chen, Y. Tian, R. Kotcher, and P. Tague, "OAuth Demystified for Mobile Application Developers," *Microsoft Res.*, Nov. 2014.
- [10] R. S. Pressman, *Software Engineering: a practitioner's approach Seventh Edition*, 7th ed. New York: The McGraw - Hill Companies, Inc, 2010.
- [11] D. Wells, "Extreme Programming: A gentle introduction." [Online]. Available: <http://www.extremeprogramming.org/>. [Accessed: 20-Feb-2019].
- [12] Y. G. Panjaitan, A. Susanto, W. -, and I. Utomo W.M, "Enkriptor-Dekriptor Isi E-Mail Berbasis Android Dengan Algoritma Blowfish," *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 8, no. 1, pp. 193–200, 2019.
- [13] A. T. S. Christanto and R. Kurniawati, "Penerapan Service Oriented Architecture Menggunakan Web Service Pada Aplikasi Perpustakaan Berbasis Android," *J. Buana Inform.*, vol. 7, no. 1, Jan. 2016.
- [14] P. R. ANDY, "Implementasi Webservice untuk Sinkronisasi Data Transkrip Nilai di Tata Usaha Fakultas Universitas Dian Nuswantoro," *Skripsi, Fakultas Ilmu Komput.*, 2016.
- [15] OAuth Community Site, "OAuth 2.0," *OAuth Community Site*, 2017. [Online]. Available: <https://oauth.net/>.
- [16] M. Jones; D. Hardt, "The OAuth 2.0 Authorization Framework: Bearer Token Usage," pp. 1–18, 2012.
- [17] D. Hardt, "The OAuth 2.0 Authorization Framework," pp. 1–76, 2012.