

Sistem Smart Parking Terdistribusi dengan Menggunakan Mobile Cloud Computing

¹M. Arfan, ²Sumardi, ³Munawar Agus Riyadi

^{1,2,3}Departemen Teknik Elektro, Universitas Diponegoro Semarang
E-mail : ¹arfan@ft.undip.ac.id, ²sumardi.undip@gmail.com , ³munawar@elektro.undip.ac.id

Abstrak

Ketersediaan lahan parkir untuk wilayah perkotaan sangat terbatas. Kantong parkir di kompleks perkantoran dan pusat keramaian masyarakat ini masih belum mampu menampung kendaraan bermotor yang terus meningkat. Menurut data terbaru Badan Pusat Statistik, kendaraan bermotor telah mencapai 129,2 juta unit pada tahun 2016. Hal ini belum didukung kemampuan pengelola lahan parkir untuk mengimbangi pertumbuhan jumlah kendaraan yang menggunakan layanan parkir. Beberapa pengelola lahan parkir telah memanfaatkan komputer sebagai alat pendukung pengelolaan parkir. Komputer difungsikan sebagai mesin pencatat data arus keluar masuk, sistem pembayaran tarif parkir dan kontrol keamanan kendaraan. Namun disisi lain informasi mengenai ketersediaan lahan parkir sangat dibutuhkan oleh pengguna jasa parkir. Data yang diberikan oleh operator parkir belum mampu memberikan informasi kepadatan lahan parkir dalam beberapa lokasi parkir dalam satu wilayah. Dari masalah tersebut, maka perlu dibangun sistem smart parking yang dapat memberikan informasi kepadatan beberapa lokasi parkir. Data pada mesin pencatat parkir dapat diolah menjadi informasi parkir pada perangkat mobile cloud kemudian memberikan rekomendasi rute terpendek dengan menggunakan algoritma Dijkstra. Dengan teknologi komputasi mobile, informasi ketersediaan lahan parkir akan berlangsung real-time. Identifikasi daya tampung dan kepadatan masing-masing kantong parkir diwujudkan dalam bentuk aplikasi berbasis mobile cloud.

Kata kunci : mobile computing, cloud computing, smart parking, Dijkstra

Abstract

The availability of parking space for urban areas is very limited. Parking lots in office complexes and community centers are still not able to accommodate the number of motor vehicles that increase each year. According data from the Central Bureau of Statistics, the number of motor vehicles has reached 129.2 million units by 2016. This has not been supported by parking managers' ability to compensate for the growing number of vehicles using parking services. Some parking managers have been using the computer as a means of parking management support. The computer functions as an incoming data logging machine, a parking rate payment system and a security control of the document. On the other hand, information about the availability of parking lots is needed by parking service users. The data provided by the parking operator has not been able to provide information on the density of the parking lot in several parking locations in one area. From the problem, it is necessary to build a smart parking system that can provide information density of several parking locations. The data on the parking register engine can be processed into parking information on mobile cloud devices then provide the shortest route recommendations using the Dijkstra algorithm. With mobile computing technology, parking lava availability information will take place in real-time. Identification of capacity and density of each parking space is realized in the form of mobile cloud-based applications.

Keywords : mobile computing, cloud computing, smart parking, Dijkstra

I. PENDAHULUAN

Kantong parkir di kompleks perkantoran dan pusat keramaian masyarakat belum mampu menampung kendaraan bermotor yang terus meningkat. Menurut data terbaru Badan Pusat Statistik, pertumbuhan jumlah kendaraan bermotor telah mencapai 129,2 juta unit pada

tahun 2016. Hal ini belum didukung kemampuan pengelola lahan parkir untuk mengimbangi pertumbuhan jumlah kendaraan yang menggunakan layanan parkir.

Beberapa pengelola lahan parkir telah memanfaatkan komputer sebagai alat pendukung pengelolaan parkir. Komputer difungsikan sebagai mesin pencatat data arus keluar masuk dan

sistem kontrol keamanan kendaraan. Kendaraan yang masuk akan memperoleh karcis yang akan digunakan kembali pada saat keluar dari kantong parkir. Selain itu karcis digunakan untuk menghitung besaran tarif parkir berdasarkan selisih waktu kendaran masuk dan keluar dari lokasi parkir, sehingga memudahkan pemilik parkir dalam mengolah pemasukan dari jasa parkir[1]. Pengguna jasa parkir seringkali kesulitan dalam mencari lokasi parkir dalam satu wilayah pusat keramaian, berpindah beberapa kantong parkir demi mendapat tempat yang kosong. Jika satu lokasi penuh, operator hanya memberi informasi lokasi kantong parkir yang lain dan tidak menjamin ketersediaan tempat parkir di lokasi tersebut. Data yang diberikan oleh operator parkir belum mampu memberikan informasi kepadatan lahan parkir dalam beberapa lokasi parkir dalam satu wilayah.

Penelitian mengenai sistem pengelolaan parkir pada umumnya menggunakan mikrokontroller untuk menghitung lama dan biaya parkir sebagai kebutuhan dasar [1]. Pada penelitian yang lain juga digunakan kamera untuk menangkap citra kendaraan pada lahan parkir [4]. Dalam penelitian juga dijelaskan mengenai teknologi cloud computing dalam memantau beberapa lahan parkir[8][9]. Disisi lain optimalisasi rute menuju kantor parkir juga dibutuhkan untuk mencari rute terdekat[7].

Dari masalah tersebut, maka perlu dibangun sistem smart parking terdistribusi yang dapat memberikan informasi kepadatan beberapa lokasi parkir di kawasan Perkotaan. Data pada mesin pencatat parkir dapat diolah menjadi informasi parkir pada perangkat *mobile cloud* kemudian memberikan rekomendasi rute terpendek dengan menggunakan algoritma Dijkstra. Dengan teknologi komputasi *mobile*, informasi ketersediaan lahan parkir akan berlangsung real-time[2]. Hal ini juga didukung oleh peningkatan jumlah pengguna teknologi *mobile cloud*. Dengan perangkat ini, komputasi pada mesin pencatat kendaraan pengelola parkir akan melakukan sinkronisasi data pengguna *mobile* secara real-time [3].

II. METODE PENELITIAN

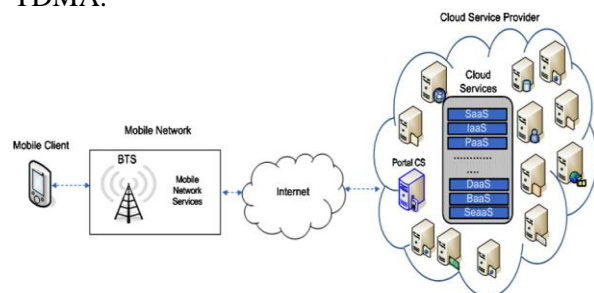
2.1. Mobile Cloud Computing

Mobile Cloud Computing merupakan komputasi berbasis *cloud computing* dengan adaptasi komunikasi bergerak. Perangkat ini menggunakan jaringan terdistribusi pada sistem *cloud*. Sistem dikembangkan menjadi tiga model

yaitu komunikasi bergerak, *wireless* dan *cloud computing*. Dengan model tersebut maka perangkat yang digunakan akan lebih mudah dalam implementasi [4].

Mobile Cloud Computing memungkinkan pengembangan aplikasi pada perangkat mobile mengembangkan kapasitas pengelolaan data dengan menambahkan keunggulan pengelolaan data secara terdistribusi. *Mobile Cloud Computing* dapat didefinisikan sebagai teknologi komputasi *mobile* yang memanfaatkan sumber daya perangkat dan jaringan yang terpadu sehingga keterbatasan penyimpanan dan mobilitas yang tinggi akan terpenuhi dengan sistem *mobile cloud* yang mampu melayani banyak perangkat *mobile* di mana pun, kapan saja melalui internet dengan mengabaikan faktor lingkungan yang heterogen dan *platform* yang berbeda. *Smartphone*, *tablet*, perangkat genggam, dan perangkat komputasi dapat dipakai adalah bagian dari kelompok ketiga sumber daya berbasis *cloud* yang entitas komputasi *mobile*[5].

Komputasi bergerak dilakukan melalui perangkat komunikasi bergerak (telepon seluler), saat ini menggunakan sistem Global System for Mobile Communications (GSM) dan CDMA. GSM merupakan standar yang pertama di gunakan di Eropa, untuk meningkatkan interoperabilitas antar perangkat seluler. Perkembangan berikutnya, teknologi dengan perangkat bergerak akhirnya digunakan diseluruh dunia sehingga terdapat kurang lebih 80 operator jaringan GSM telah menjalankan teknologi tersebut. Teknologi ini mulai berjalan pada frekuensi 900 dan 1800 MHZ di seluruh Eropa dan Inggris. Di beberapa negara bagian Amerika juga menggunakan frekuensi 1900 MHZ berbasis TDMA.



Gambar 1. Infrastruktur Mobile Cloud Computing

Aplikasi dijalankan pada server penyedia layanan kemudian dikirim ke pengguna melalui transaksi *mobile browser*. Perkembangan aplikasi *mobile browser* yang mendukung berbagai platform pada setiap ponsel pintar menjadikan

komputasi mobile mengalami peningkatan pada sisi jumlah pengguna ponsel dan dimanfaatkan oleh pengelola parkir dengan teknologi *smart parking*[3].

Dengan komputasi mobile *cloud* pengguna sistem tidak perlu menyiapkan server tetapi hanya menyiapkan alat akses berupa *smartphone* yang memiliki akses internet. Semua transaksi data akan diolah *cloud server* dan hasilnya dapat diakses melalui *smartphone* sehingga memudahkan pengguna. Komunikasi data antara *server* dan *smartphone* menggunakan jaringan *mobile* yang terkoneksi layanan *cloud*. Dengan menggunakan sistem berbasis *cloud* akan terlihat log pengiriman data dan segala kemungkinan kebocoran data pada jaringan *cloud* yang dirancang. Dalam perancangan sistem parkir akan ditemukan beberapa metode penyajian yang tepat berdasarkan literature dan kondisi dinamis di lapangan kemudian dari metode tersebut dilakukan desain representasi kebutuhan. Perancangan sistem akan menghasilkan model dan desain yang bertujuan untuk memudahkan implementasi sistem komputasi berbasis *mobile cloud* [6].

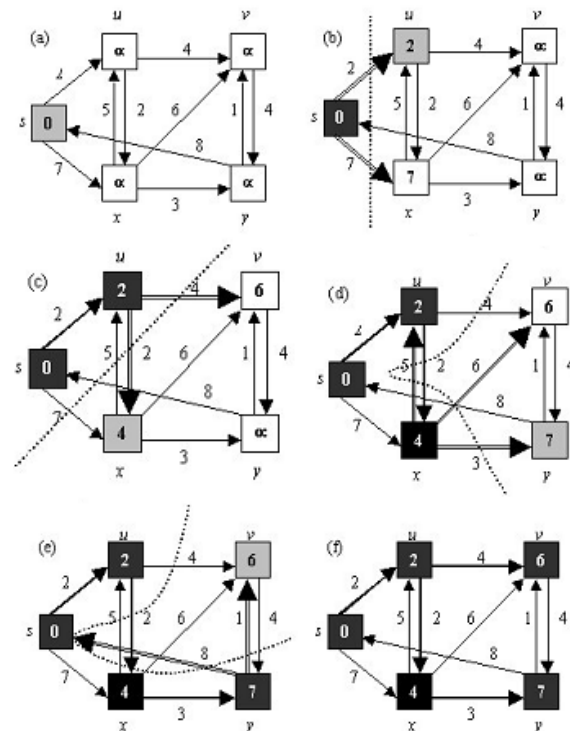
2.2. Algoritma Dijkstra

Algoritma Dijkstra merupakan algoritma untuk menentukan rute terpendek. Rute dimulai dari simpul awal ke titik target dalam grafik. Algoritma berbentuk pohon untuk menentukan jalur terpendek dari titik awal, sumber, ke semua titik lain dalam grafik. Algoritma Dijkstra dapat dituliskan $G = (V, E)$, di mana V merupakan simpul dan E adalah ujung. Algoritma Dijkstra mensyaratkan bahwa jarak antara node yang memiliki tepi dan memiliki nilai. Jika ingin menemukan jalur terpendek dari s ke v [7].

Pertama kali algoritma Dijkstra akan menginisialisasi $d(s) = 0$ karena jarak tempuh nol untuk pergi dari s ke s , dan menginisialisasi semua $d(i) = \text{infinity}$ karena belum ada jalur yang ditentukan,. Setelah itu akan dilakukan pendefinisian potongan (garis putus-putus) menjadi garis yang membagi simpul-simpul yang menjadi garis yang membagi simpul-simpul yang jarak terpendek $d(i)$ yang telah ditentukan yang digambarkan dengan simpul warna hitam. Sedangkan titik abu-abu dari jarak terpendek adalah yang ditentukan saat ini [7].

Algoritma Dijkstra terdiri dari Titik s sebagai titik awal dan titik y sebagai titik tujuan. Kemudian akan ditentukan rute yang harus diambil dan memiliki nilai jarak paling dekat. Untuk memperoleh rute terdekkan, maka akan ditambahkan beberapa kotak sebagai representasi beberapa label. Pada gambar 2(a), titik dimulai dari s , dan mempertimbangkan semua sisi yang

menuju ke- s dimana diperoleh $d(s)=0$. Kemudian jarak terdekat ke titik s ditandai dengan warna hitam. Pada gambar 2(b), sudah dibuatkan potongan dengan garis putus-putus dan grafik masih mempertimbangkan semua sisi yang memotong, yaitu, dua sisi yang memiliki panjang 2 (s,u) dan 7 (s,x), dan dua baris. Setelah itu terpilih yang terpendek yaitu, (s,u). Terlihat nilai $d(u)$ di u , yang tidak terbatas dan jelas lebih besar dari 2.



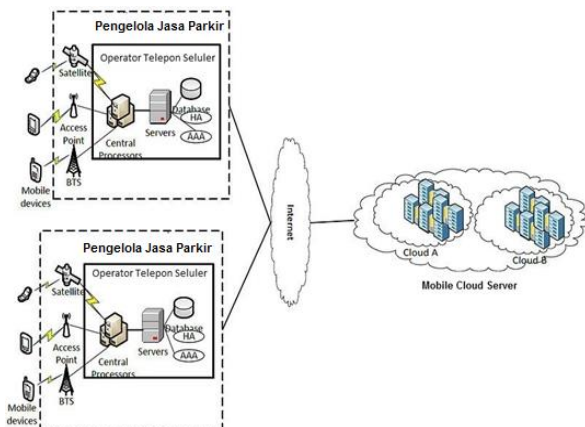
Gambar 2. Algoritma Dijkstra

Oleh karena itu, *infinity* diganti dengan 2 sebagai nilai $d(u)$. Jadi jarak terpendek juga telah ditentukan [8]. Dengan mempertimbangkan semua sisi yang memotong simpul maka pemotongan dilakukan antara s,u , dan sisa dari simpul. Grafik memiliki panjang 4 (u,v) dan 2 (u,x) masing-masing 6 lebih kecil dari *infinity*, oleh karena itu diganti. Terlihat 4 lebih kecil dari 7 dan nilai $d(x)$ saat ini, yang juga akan diganti. Karena (u,x) lebih pendek dari (u,y), kemudian dipilih untuk berada pada sisi terpendek, dan jalur terpendek x kembali ditentukan. Sisi ujung yang sedikit gelap mewakili ujung-ujung yang telah dipilih sebagai daftar tepi di jalur terpendek. Proses ini berlanjut hingga jarak terpendek dari semua node telah ditemukan sebelumnya. Dengan menggunakan algoritma Dijkstra, maka prose untuk menemukan jalur terpendek akan terus berlanjut dan berhenti jika jalur terpendek sudah ditentukan.

III. HASIL DAN PEMBAHASAN

3.1. Infrastruktur Smart Parking

Sistem *Smart Parking* dibangun dengan menggunakan *server* dengan teknologi mesin *server* virtual. Mesin *server* utama digunakan sebagai *server* basis data dan mesin *server* cadangan sebagai *server* backup akan diimplementasikan dalam satu *server* yang telah mendukung teknologi Kernel-based Virtual Machine (KVM) untuk virtualisasi mesin *server* dan satu unit yang lain sebagai *web server* menangani aplikasi berbasis *web*.



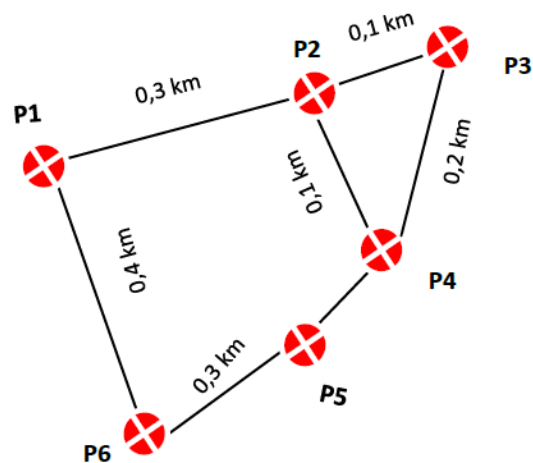
Gambar 3. Infrastruktur *Smart Parking Mobile Cloud Computing*

Dengan komputasi *mobile cloud* pengguna sistem tidak perlu menyiapkan *server* tetapi hanya menyiapkan alat akses berupa *smartphone* yang memiliki akses internet. Semua transaksi data akan diolah *cloud server* dan hasilnya dapat diakses melalui *smartphone*. Dari sistem ini akan terlihat kemungkinan kebocoran data pada jaringan *cloud* yang dirancang.[9]

3.2. Sistem Pemetaan Lahan Parkir

Perancangan sistem dilakukan dengan melakukan pemetaan kawasan perkantoran untuk membuat jalur yang akan dilalui kendaraan kemudian dituangkan dalam bentuk algoritma. Jarak antara kantong parkir akan dihitung untuk memberikan rekomendasi kepada pengguna parkir.

Dari data tersebut akan diperoleh jumlah kantong parkir yang kemudian disajikan dalam bentuk algoritma Dijkstra untuk menghitung jarak rekomendasi antar masing-masing kantong parkir. Informasi tersebut akan memberikan jalur alternatif untuk pengguna parkir. Kendaraan masuk ke kawasan parkir akan mengirim permintaan parkir kepada mesin *server*. *Server* kemudian melakukan sinkronisasi dengan kantong parkir yang ada tersedia.



Gambar 4. Jarak Antar Lahan Parkir

Kemudian *server* akan memberikan notifikasi dan diberikan informasi mengenai lokasi parkir yang tersedia. Proses ini akan diberi batasan waktu untuk menghindari permintaan palsu yang akan menyulitkan pengguna jasa parkir lain [10].

Dengan menghitung jarak terdekat dengan algoritma Dijkstra maka pengguna jasa parkir mendapatkan rekomendasi lokasi parkir terdekat dan sedang kosong.



Gambar 5. Rekomendasi Kantong Parkir

IV. KESIMPULAN

Teknologi *mobile cloud* untuk pengelolaan parkir mampu memberikan pelaporan kondisi beberapa kantong parkir secara *real-time* sehingga pengguna parkir mendapat kemudahan mencari lokasi parkir yang tersedia. Pengembangan sistem *smart parking* diharapkan pada modul pengenalan jumlah kendaraan berdasarkan ukuran sehingga ketersediaan parkir lebih optimal.

DAFTAR PUSTAKA

- [1] A. Anindita, Sudjadi, Darjat, "Sistem Informasi Area Parkir Berbasis Mikrokontroler Atmega 16," *Transmisi*, vol. 15, no 3, pp. 132–137, 2013.

- [2] T. N. A. M. Pham, M. Tsai, and D. U. C. B. Nguyen, "A Cloud-Based Smart-Parking System Based on Internet-of-Things Technologies," pp. 1581–1591, 2015.
- [3] J. Rico, J. Sancho, B. Cendon, and M. Camus, "Parking easier by using context information of a smart city: Enabling fast search and management of parking resources," Proc. - 27th Int. Conf. Adv. Inf. Netw. Appl. Work. WAINA 2013, pp. 1380–1385, 2013.
- [4] M. Heimberger, J. Horgan, C. Hughes, J. McDonald, and S. Yogamani, "Computer vision in automated parking systems: Design, implementation and challenges," Image Vis. Comput., vol. 68, pp. 88–101, 2017.
- [5] A. N. Khan, M. L. Mat Kiah, S. U. Khan, and S. A. Madani, "Towards secure mobile cloud computing: A survey," Futur. Gener. Comput. Syst., no. c, 2012.
- [6] M. L. Lackner and A. Panholzer, "Parking functions for mappings," J. Comb. Theory. Ser. A, vol. 142, pp. 1–28, 2016.
- [7] L. M. S. Bento, D. R. Boccardo, R. C. S. Machado, F. K. Miyazawa, V. G. Pereira de Sá, and J. L. Szwarcfiter, "Dijkstra graphs," Discret. Appl. Math., 2017.
- [8] Yujin and G. Xiaoxue, "Optimal Route Planning of Parking Lot Based on Dijkstra Algorithm," Proc. - 2017 Int. Conf. Robot. Intell. Syst. ICRIS 2017, pp. 221–224, 2017.
- [9] T. Clohessy, T. Acton, and L. Morgan, "Smart city as a service (SCaaS): A future roadmap for e-government smart city cloud computing initiatives," Proc. - 2014 IEEE/ACM 7th Int. Conf. Util. Cloud Comput. UCC 2014, pp. 836–841, 2014.
- [10] N. Chen, L. Wang, L. Jia, H. Dong, and H. Li, "Parking Survey Made Efficient in Intelligent Parking Systems," Procedia Eng., vol. 137, pp. 487–495, 2016.