

Pemanfaatan Algoritma Greedy Pada Aplikasi Pemesanan Jasa Grooming Kucing Berbasis Android

Fachri Ikhwani^{*1}, Hendra Kurniawan²

^{1,2}Jurusan Teknik Informatika; STMIK Pontianak. Jl. Merdeka No.372 Pontianak, 0561-735555
e-mail: ^{*1}Fachri.Ikhwani@gmail.com, ²Hendra.kurniawan@stmikpontianak.ac.id

Abstrak

Penelitian ini yang berjudul Pemanfaatan Algoritma Greedy Pada Aplikasi Pemesanan Jasa Grooming Kucing Berbasis Android ini dibuat agar mempermudah pemilik kucing untuk memesan jasa grooming untuk kucing mereka. Grooming merupakan bagian dari merawat diri makhluk hidup, baik manusia maupun hewan bisa melakukan grooming untuk menjaga penampilannya. Tujuan dari grooming untuk kucing sendiri adalah membersihkan kucing dari segala bentuk bakteri ataupun kutu kucing yang sangat berbahaya, dan juga menjaga penampilan agar terlihat bersih dan lucu. Metode perancangan perangkat lunak menggunakan model RAD. Bentuk penelitian yang digunakan dalam penelitian ini adalah studi literatur dengan menggunakan metode penelitian research and development. Metode pengumpulan data berupa pengumpulan data primer dan sekunder yang didapat dari hasil penyebaran kuesioner, observasi dan studi dokumentasi. Pemodelan perangkat lunak menggunakan menggunakan model UML (Unified Modelling Language). Algoritma yang digunakan dalam perangkat lunak ini adalah algoritma greedy, perancangan perangkat lunak pemesanan jasa grooming kucing ini dibuat dengan menggunakan Android Studio dengan database Firebase. Metode pengujian yang digunakan adalah metode pengujian black box. Hasil dari penelitian ini adalah perangkat lunak pemesanan jasa grooming kucing menggunakan algoritma greedy yang dapat digunakan untuk mempermudah pemilik kucing untuk memesan jasa grooming untuk kucing mereka.

Kata kunci— Greedy, Firebase, RAD, Android Studio

Abstract

This research is titled Utilization of Greedy Algorithm on Applications for Android-based Cat Grooming Services is made to make it easier for cat owners to order grooming services for their cats. Grooming is part of caring for a living creature, both humans and animals can do grooming to maintain their appearance. The goal of grooming for cats themselves is to clean cats from all forms of bacteria or cat lice which are very dangerous, and also maintain the appearance to look clean and cute. The software design method uses the RAD model. The form of research used in this study is literature study using research and development research methods. Data collection methods in the form of primary and secondary data collection obtained from the results of questionnaires, observation and documentation studies. Software modeling uses the UML (Unified Modeling Language) model. The algorithm used in this software is the greedy algorithm, the design of the cat grooming service ordering software is made using Android Studio with the Firebase database. The testing method used is the black box testing method. The results of this study are software ordering cat grooming services using greedy algorithms that can be used to make it easier for cat owners to order grooming services for their cats.

Keywords— Greedy, Firebase, RAD, Android Studio

1. PENDAHULUAN

Jasa grooming sendiri merupakan bagian dari merawat diri makhluk hidup, baik manusia maupun hewan bisa melakukan grooming untuk menjaga penampilannya. Tujuan dari grooming

untuk kucing sendiri adalah membersihkan kucing dari segala bentuk bakteri ataupun kutu kucing yang sangat berbahaya, dan juga menjaga penampilan agar terlihat bersih dan lucu.

Aplikasi pemesanan jasa *grooming* kucing berbasis mobile ini menggunakan algoritma *greedy* untuk pencarian jalur terpendek pada saat menentukan jalur mana yang akan dilewati oleh *groomer* agar bisa memberikan jalur tercepat dan terpendek dari tempat asal menuju ketempat yang akan dituju dengan cepat. Pengoptimalan kinerja suatu sistem, baik untuk meminimalkan biaya atau mempercepat jalannya suatu proses.

Algoritma Greedy merupakan metode yang paling populer untuk memecahkan persoalan optimasi. Hanya ada dua macam persoalan optimasi, yaitu maksimasi dan minimasi. Algoritma *greedy* membentuk solusi langkah per langkah[1]. Pada setiap langkah, terdapat banyak pilihan yang perlu dieksplorasi. Oleh karena itu, pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihan. Pada setiap langkahnya merupakan pilihan, untuk membuat langkah optimum lokal dengan harapan bahwa langkah sisanya mengarah ke solusi optimum global. Sesuai arti harafiah, Greedy berarti tamak. Prinsip utama dari algoritma ini adalah mengambil sebanyak mungkin apa yang dapat diperoleh sekarang. Algoritma Greedy dapat menentukan sebuah jalur terpendek antara node-node yang akan digunakan dengan mengambil secara terus-menerus dan menambahkannya ke dalam jalur yang akan dilewati. Atau pada notasi big-O dituliskan $O(n^2 \log^2(n))$.

Dalam membentuk solusi, algoritma *greedy* pada setiap langkahnya akan mengambil pilihan yang merupakan optimum lokal atau pilihan yang sesuai dengan spesifikasi pembuat algoritma. Dengan pengambilan pilihan yang sesuai pada setiap langkah ini diharapkan solusi yang didapat optimum global atau sesuai keinginan pembuat algoritma. Pada setiap langkah algoritma *greedy*, kita akan mendapat optimum lokal. Bila algoritma berakhir maka diharapkan optimum lokal ini akan menjadi optimum global. Sehingga sebenarnya algoritma *greedy* mengasumsikan bahwa optimum lokal ini merupakan bagian dari optimum global.

Pada penelitian tentang algoritma *greedy* sudah banyak dilakukan sebelumnya, Algoritma Greedy adalah algoritma yang memecahkan masalah langkah demi langkah dan merupakan salah satu metode dalam masalah optimasi. Pendekatan yang digunakan di dalam algoritma Greedy adalah membuat pilihan yang terlihat memberikan perolehan terbaik yaitu dengan membuat pilihan optimum lokal pada setiap langkah dan diharapkan akan mendapatkan solution optimum global[1]. Aplikasi dengan algoritma *greedy* akan menghitung lokasi yang paling optimal untuk dikunjungi dengan waktu yang dimiliki pengguna[2]. Hasil dari algoritma *greedy* dan dibanding dengan algoritma lain, algoritma *greedy* memiliki hasil optimasi yang lebih bagus dan tentu saja dengan waktu komputasi yang lebih cepat[3]. Algoritma Greedy dapat menentukan jalur mana yang akan diambil terlebih dahulu atau dapat disebut dengan jalur optimum lokal sehingga sampai seluruh jalur diambil pada akhir perjalanan dan menciptakan rute perjalanan terpendek[4]. Keberhasilan algoritma Greedy dalam penanganan kasus pemilihan rute terpendek juga bergantung pada pemilihan pengurutan simpul yang dipakai saat penelusuran terjadi. Akan tetapi, penggunaan algoritma Greedy tetap membantu dalam mengurangi rute tempuh suatu tujuan dan memberikan sebuah solusi optimal lokal, dimana solusi tersebut cukup efektif untuk menentukan rute tempuh terdekat dalam pencarian sebuah rumah sakit[5].

2. METODE PENELITIAN

Bentuk penelitian yang digunakan dalam penelitian ini adalah studi literatur (*literature review*). Studi literatur merupakan studi yang bisa digunakan sebagai bahan untuk mengumpulkan dan mengkaji data dengan membaca berbagai literatur seperti buku, skripsi, jurnal maupun bentuk tulisan lainnya yang isinya berkaitan erat dengan masalah yang akan diteliti sebagai bahan referensi.

Metode yang digunakan dalam penelitian ini adalah metode penelitian *Research and Development (R&D)*. “Metode penelitian *Research and Development* merupakan hasil dari penelitian pengembangan, tidak mengembangkan suatu produk baru atau menyempurnakan produk yang telah ada, melainkan untuk menemukan pengetahuan atau jawaban atas permasalahan praktis”[6]. Metode penelitian dan pengembangan juga didefinisikan sebagai sesuatu metode penelitian yang digunakan untuk menghasilkan produk tertentu, dan menguji keefektifan produk tersebut. Tujuan menggunakan metode ini yaitu untuk menghasilkan sebuah perangkat lunak pemesanan jasa grooming kucing berbasis android dengan algoritma *greedy*.

Metode pengumpulan data merupakan bagian paling penting dalam sebuah penelitian. Ketersediaan data akan sangat menentukan dalam proses pengolahan dan analisa selanjutnya. Karenanya, dalam pengumpulan data harus dilakukan teknik yang menjamin bahwa data diperoleh itu benar, akurat dan bisa dipertanggungjawabkan sehingga hasil pengolahan dan analisa data tidak biasa. Metode yang digunakan peneliti dalam melakukan pengumpulan data adalah studi dokumentasi, yaitu peneliti mengumpulkan serta mempelajari bahan-bahan tertulis yang berhubungan dengan penggunaan algoritma *greedy* yang di dapat melalui artikel, buku, e-book dan pencarian di internet terhadap materi algoritma *greedy*.

3. HASIL DAN PEMBAHASAN

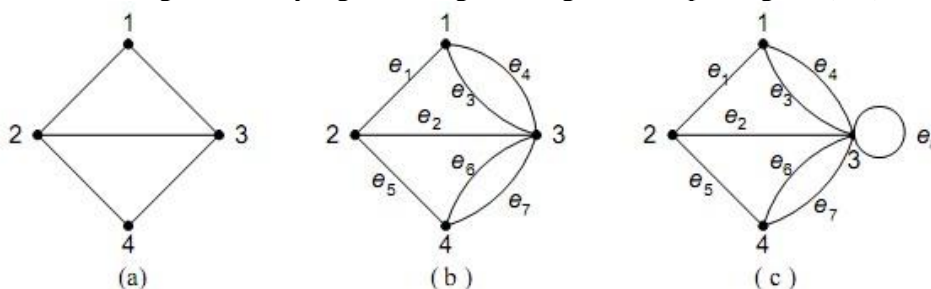
Algoritma *greedy* adalah algoritma yang memecahkan masalah langkah demi langkah, pada setiap langkah. Mengambil pilihan yang terbaik yang dapat diperoleh saat itu, berharap bahwa dengan memilih optimum local pada setiap langkah akan mencapai optimum global. Algoritma *greedy* mengasumsikan bahwa optimum lokal merupakan bagian dari optimum global. Persoalan optimasi dalam konteks algoritma *greedy* disusun oleh elemen-elemen sebagai berikut: Himpunan kandidat, C . Himpunan ini berisi elemen-elemen pembentuk solusi. Pada setiap langkah, satu buah kandidat diambil dari himpunannya. Himpunan solusi, S . Merupakan himpunan dari kandidat-kandidat yang terpilih sebagai solusi persoalan. Himpunan solusi adalah himpunan bagian dari himpunan kandidat. Fungsi seleksi – dinyatakan sebagai predikat SELEKSI – merupakan fungsi yang pada setiap langkah memilih kandidat yang paling mungkin untuk mendapatkan solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya. Fungsi kelayakan (feasible) – dinyatakan dengan predikat LAYAK – merupakan fungsi yang memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar kendala yang ada. Fungsi obyektif, merupakan fungsi yang memaksimalkan atau meminimumkan nilai solusi. Kita berharap optimum global merupakan solusi optimum dari persoalan. Namun, ada kalanya 2 optimum global belum tentu merupakan solusi optimum (terbaik), tetapi dapat merupakan solusi sub-optimum atau pseudo-optimum.

Hal ini dapat dijelaskan dari dua faktor berikut, algoritma *greedy* tidak beroperasi secara menyeluruh terhadap semua alternatif solusi yang ada. Pemilihan fungsi SELEKSI: fungsi SELEKSI biasanya didasarkan pada fungsi obyektif (fungsi SELEKSI bisa saja identik dengan fungsi obyektif). Jika fungsi SELEKSI tidak identik dengan fungsi obyektif, kita harus memilih fungsi yang tepat untuk menghasilkan nilai yang optimum. Karena itu, pada sebagian masalah algoritma *greedy* tidak selalu berhasil memberikan solusi yang benarbenar optimum. Tetapi, algoritma *greedy* pasti memberikan solusi yang mendekati (approximation) nilai optimum. (Defindal dkk.,) Algoritma *greedy* untuk mencari lintasan terpendek dapat dirumuskan sebagai berikut, Periksa semua sisi yang langsung bersisian dengan simpul a . Pilih sisi yang bobotnya terkecil. Sisi ini menjadi lintasan terpendek pertama, sebut saja $L(1)$. Tentukan lintasan terpendek kedua dengan cara berikut: i. hitung: $d(i) = \text{panjang } L(1) + \text{bobot sisi dari simpul akhir } L(1) \text{ ke simpul } i$ yang lain ii. pilih $d(i)$ yang terkecil Bandingkan $d(i)$ dengan bobot sisi (a, i) . Jika

bobot sisi (a, i) lebih kecil daripada $d(i)$, maka $L(2) = L(1) \cup$ (sisi dari simpul akhir $L(i)$ ke simpul i). Dengan cara yang sama, ulangi langkah 2 untuk menentukan lintasan terpendek berikutnya.

Graf suatu cabang ilmu yang memiliki banyak terapan. Banyak sekali struktur yang bisa direpresentasikan dengan graf, dan banyak masalah yang bisa diselesaikan dengan bantuan graf. Seringkali graf digunakan untuk merepresentasikan suatu jaringan. Misalkan jaringan jalan raya dimodelkan graf dengan kota sebagai simpul (vertex/node) dan jalan yang menghubungkan setiap kotanya sebagai sisi (edge) yang bobotnya (weight) adalah panjang dari jalan tersebut. Dalam beberapa model persoalan dimungkinkan bahwa bobot dari suatu sisi bernilai negatif. Misalkan simpul merepresentasikan kota, sisi merepresentasikan perjalanan yang memungkinkan, dan bobot dari setiap sisi adalah biaya yang dikeluarkan dalam perjalanan yang memungkinkan, dan bobot dari setiap sisi adalah jarak yang ditempuh dalam perjalanan tersebut. Graf G didefinisikan sebagai pasangan himpunan (V, E) . ditulis dengan notasi $G = (V, E)$, yang dalam hal ini V adalah himpunan tidak kosong dari simpul-simpul (vertices atau node) dan E adalah himpunan sisi (edges atau arcs) yang menghubungkan sepasang simpul. V tidak boleh kosong, sedangkan E boleh kosong. (Ristono dan Puryani, 2011).

Simpul pada graf dapat dinomori dengan huruf, seperti $a, b, c, \dots, v, w, \dots$, dengan bilangan asli $1, 2, 3, \dots$, atau gabungan keduanya. Sedangkan sisi yang menghubungkan simpul u dengan simpul v dinyatakan dengan pasangan (u, v) atau dinyatakan dengan lambing e_1, e_2, \dots . Dengan kata lain, jika e adalah sisi yang menghubungkan simpul u dengan simpul v , maka e dapat ditulis sebagai $e = (u, v)$. Secara geometri graf digambarkan sebagai sekumpulan noktah (simpul) di dalam bidang dwimitra yang dihubungkan dengan sekumpulan garis (sisi).



Gambar 1. Tiga buah graf (a) graf sederhana, (b) graf ganda, dan (c) graf semu

Graf sederhana adalah graf yang tidak mengandung gelang maupun sisi-ganda. Graf ganda adalah graf yang mengandung sisi-ganda. Graf semu adalah graf yang mengandung gelang (*loop*), graf semu lebih umum daripada graf ganda, karena sisi pada graf semu dapat terhubung ke dirinya sendiri.[9].

Persoalan mencari lintasan terpendek di dalam graf merupakan salah satu persoalan optimasi. Graf yang digunakan dalam pencarian lintasan terpendek adalah graf berbobot (weighted graph), yaitu graf yang setiap sisinya diberikan suatu nilai atau bobot. Bobot pada sisi graf dapat menyatakan jarak antar kota, waktu pengiriman pesan, ongkos pembangunan, dan sebagainya. Asumsi yang digunakan di sini adalah bahwa semua bobot bernilai positif. Lintasan terpendek adalah jalur yang dilalui dari suatu node ke node lain dengan besar atau nilai pada sisi yang jumlah akhirnya dari node awal ke node akhir paling kecil. Lintasan terpendek adalah lintasan minimum yang diperlukan untuk mencapai suatu tempat dari tempat lain. Lintasan minimum yang dimaksud dapat dicari dengan menggunakan graf. Graf yang digunakan adalah graf yang berbobot yaitu graf yang setiap sisinya diberikan suatu nilai atau bobot.

Adapun ketiga fase tersebut adalah requirements planning (perencanaan syarat-syarat), RAD design workshop (workshop desain RAD), dan implementation (implementasi)[7]. Model RAD juga merupakan suatu pendekatan berorientasi objek terhadap membangun system yang mencakup suatu metode membangun perangkat-perangkat lunak. Tujuannya adalah

mempersingkat waktu pengerjaan aplikasi serta proses yang dihasilkan didapat secara cepat dan tepat.

Pada tahap ini, dilakukan semua pengumpulan kebutuhan user yang berkaitan dengan perancangan aplikasi pemesanan jasa grooming kucing. Fase ini, pengguna dan penganalisis bertemu untuk mengidentifikasi tujuan-tujuan aplikasi atau sistem serta untuk mengidentifikasi syarat-syarat informasi yang ditimbulkan dari tujuan-tujuan tersebut, orientasi dalam fase ini adalah menyelesaikan masalah-masalah. Meskipun teknologi informasi dan sistem bisa mengarahkan sebagian dari sistem yang diajukan, fokusnya akan selalu tetap pada upaya pencapaian tujuan-tujuan. Pengembang sistem harus dapat mencari informasi untuk memenuhi kebutuhan aplikasi yang dibangun.

Aplikasi pemesanan jasa grooming kucing menggunakan algoritma greedy adalah sebuah aplikasi pemesanan yang akan membantu dalam memperlancar proses memilih rute terpendek yang telah ditentukan lokasi awal dan tujuan, Kehadiran aplikasi pemesanan jasa grooming berbasis mobile ini akan dapat membantu pemilik kucing untuk memesan grooming untuk kucingnya. Design Workshop merupakan tahapan awal untuk menganalisis sebuah kebutuhan untuk skema dasar. Berikut ini beberapa analisis yang didasarkan untuk pembangunan aplikasi.

Analisis domain pada penelitian ini adalah sistem operasi android merupakan salah satu sistem operasi smartphone yang sistem kerjanya berjalan diatas kernel linux dan menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi sendiri yang dapat digunakan oleh bermacam peranti bergerak. Karena sifat android yang open source maka aplikasi yang dibangun memiliki reliabilitas lebih tinggi dibandingkan aplikasi mobile GIS yang bersifat closed source lainnya.

Selain itu dengan membangun aplikasi pada sistem operasi ponsel android, maka kita dapat memanfaatkan fitur yang disediakan oleh Google yaitu Google Map. Sehingga kita dapat menggunakan peta virtual secara gratis, juga tidak perlu membuat server khusus untuk mengelola data peta tersebut. Aplikasi yang akan dibangun adalah berbasis mobile yang memiliki sistem operasi Android, berfungsi sebagai aplikasi navigasi. Dalam aplikasi ini user dapat menggunakan aplikasi ini untuk memesan grooming , melihat informasi kalkulasi jarak antara groomer dan user, melihat informasi groomer, dan mengetahui posisi user, sedangkan untuk groomer dapat menerima pesanan grooming, melihat rute antara groomer dan user, melihat informasi user dan mengetahui posisi groomer.

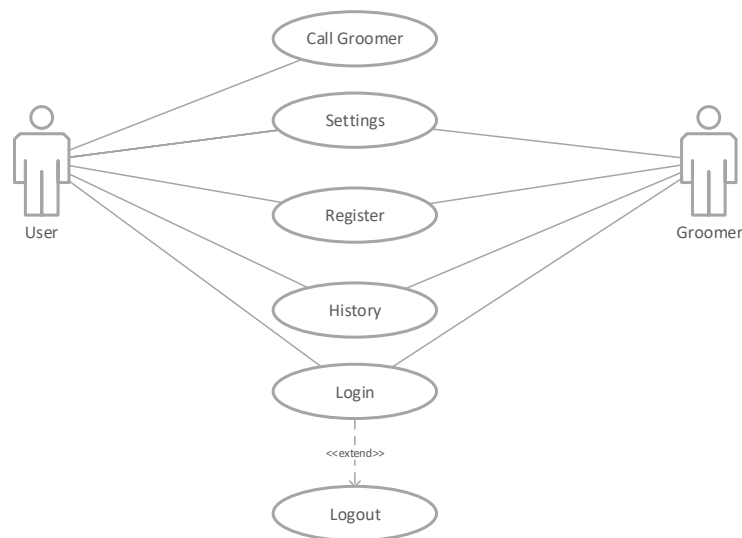
Identifikasi sistem merupakan gambaran model diagram untuk menunjukkan tata letak pada sebuah sistem secara fisik dengan menampakkan bagian-bagian perangkat lunak yang berjalan pada perangkat keras yang dibangun akan menampakkan bagian-bagian software yang berjalan pada hardware. Pada model pemrosesan aplikasi pemesanan jasa grooming kucing ini dapat merepresentasikan real world (dunia nyata) diaplikasi sebagaimana lembaran peta dapat mempresentasikan dunia nyata diatas kertas, tetapi aplikasi pemesanan ini memiliki kekuatan lebih dan fleksibilitas dan pada lembaran peta kertas. Peta merupakan representasi grafis dan dunia nyata, objek-objek yang direpresentasikan diatas peta tersebut unsur peta atau map feature contohnya adalah sungai, taman, kebun, jalan, dan lain-lain. Karena peta mengorganisasikan unsur-unsur berdasarkan lokasi-lokasinya, peta sangat baik dalam memperlihatkan hubungan atau relasi yang dimiliki oleh unsur-unsurnya.

Strategi dalam tahapan perancangan aplikasi pemesanan jasa grooming mengacu pada perancangan berbasis objek. Strategi ini dalam istilah aslinya disebut sebagai OOD (*Object Oriented Desistem Informasi Geografis*) dan dianggap menjadi strategi perencanaan paling modern. Dalam penelitian ini penulis menggunakan UML (Unified Modeling Language). UML memiliki beberapa konsep dasar yang diabstrasikan dalam bentuk *structural classification*, *dynamic behavior*, dan *model management*.

Hal terpenting dalam penggunaan UML adalah pembuatan diagram yang sesuai dengan analisis dan pengembangan sistem. Notasi-notasi UML mampu mempresentasikan rancangan sistem yang berorientasi objek sehingga menjadi lebih mudah ketika rancangan nantinya

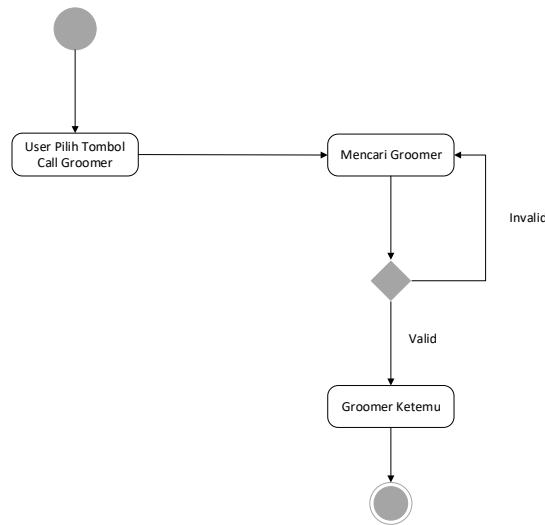
diimplementasikan pada bahasa pemrograman objek seperti java. Berikut ini adalah perancangan arsitektur perangkat lunak yang dimodelkan dengan UML :

Use case diagram menjelaskan manfaat sistem jika dilihat menurut pandangan orang yang berada diluar sistem atau actor. Diagram ini menunjukkan fungsionalitas suatu sistem atau kelas dari bagaimana sistem berinteraksi dengan dunia luar. Perancangan proses yang terjadi dalam aplikasi pemesanan jasa grooming kucing dengan Use Case Diagram sebagai berikut.



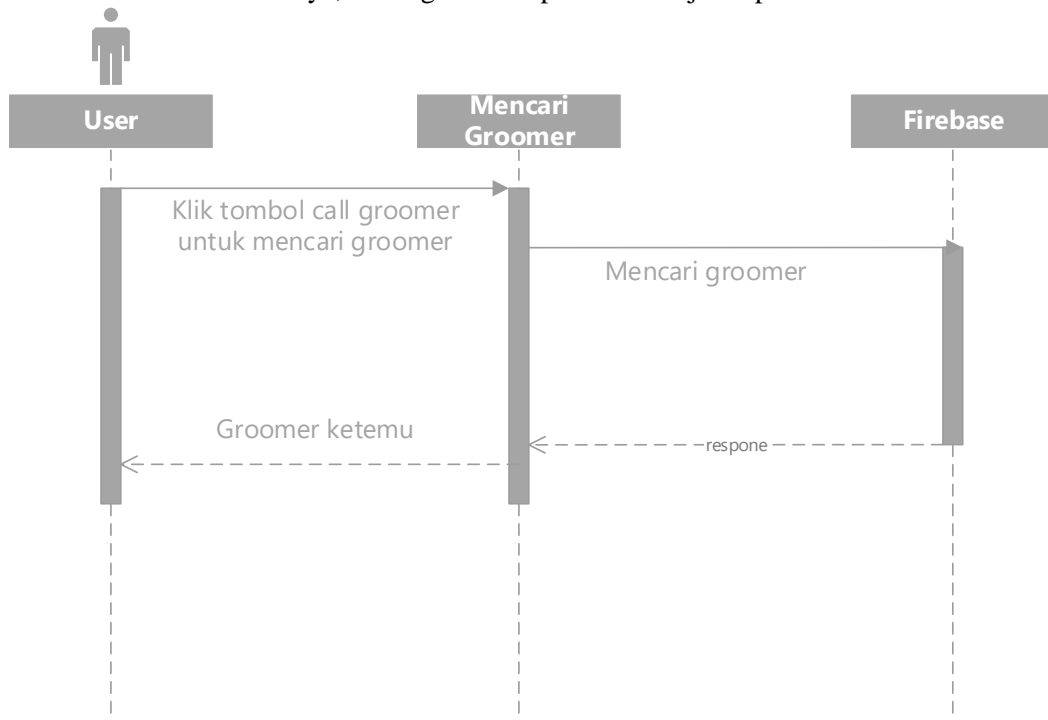
Gambar 2. Use Case Diagram

Activity Diagram menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, decision yang mungkin terjadi dan bagaimana mereka berakhir. *Activity Diagram* juga dapat menggambarkan proses parallel yang mungkin terjadi pada beberapa eksekusi. *Activity Diagram* merupakan state diagram khusus, dimana sebagian besar state adalah action dan sebagian besar transisi ditrigger oleh selesainya state sebelumnya (*internal processing*), oleh karena itu *Activity Diagram* tidak menggambarkan behavior internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Berikut adalah tampilan dari activity diagram call groomer pada aplikasi pemesanan jasa grooming kucing berbasis android :



Gambar 3. Activity Diagram Call Groomer

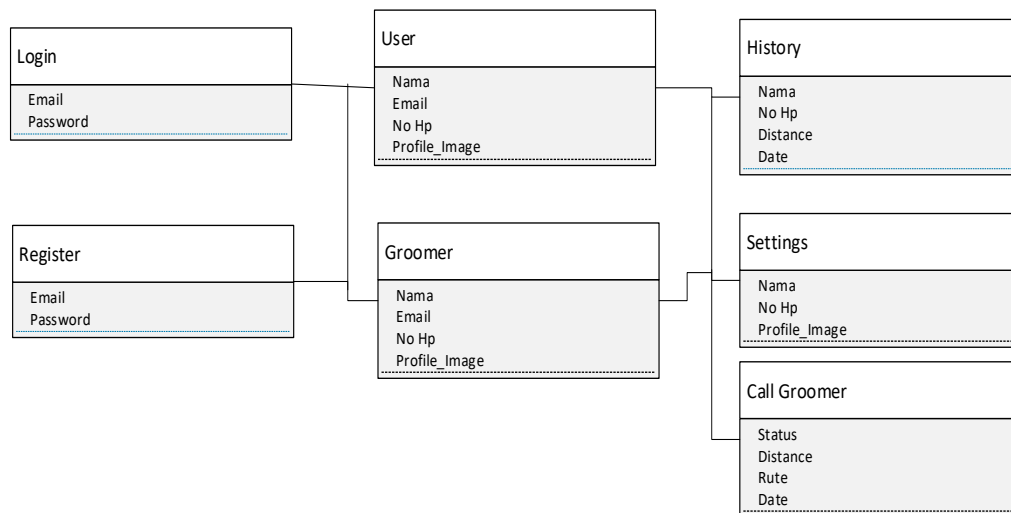
Sequence diagram menggambarkan interaksi antar objek didalam dan disekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. *Sequence Diagram* terdiri antar dimensi vertical (waktu) dan dimensi horizontal (objek-objek yang terkait). Sequence Diagram dapat digunakan untuk menggambarkan skenario atau langkah-langkah yang dilakukan sebagai respon dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivasi tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. Masing-masing objek, termasuk actor, memiliki lifeline vertical. Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, message akan dipetakan menjadi operasi/metode dari class.



Gambar 4. Sequence Diagram Call Groomer

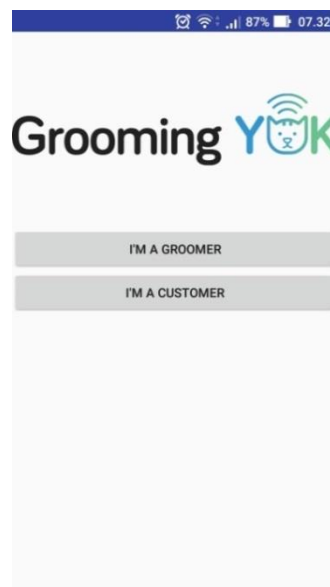
Class Diagram adalah sekelompok dari object yang mempunyai attribute, operasi, hubungan yang semantik. Sebuah class diagram mengimplementasikan satu atau lebih interface. Class Diagram dapat digambarkan sebagai sebuah persegi panjang, yang mempunyai sebuah nama, attribute, dan metoda pengoperasiannya. *Class diagram* digunakan untuk memodelkan perancangan statik dari gambaran sistem. Biasanya meliputi permodelan *vocabulary* dari sistem, permodelan kerjasama, atau permodelan skema.

Class diagram dapat digunakan untuk membangun sistem yang dapat dieksekusi melalui teknik forward and reverse, selain untuk penggambaran, penspesifikasian, dan pendokumentasian struktur model.

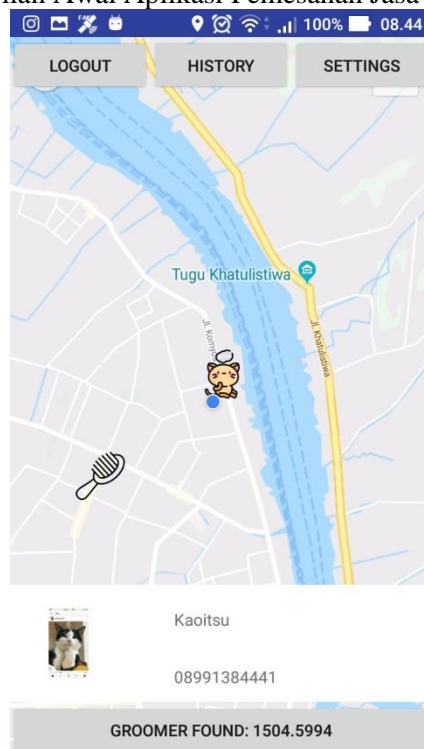


Gambar 5. Class Diagram Call Groomer

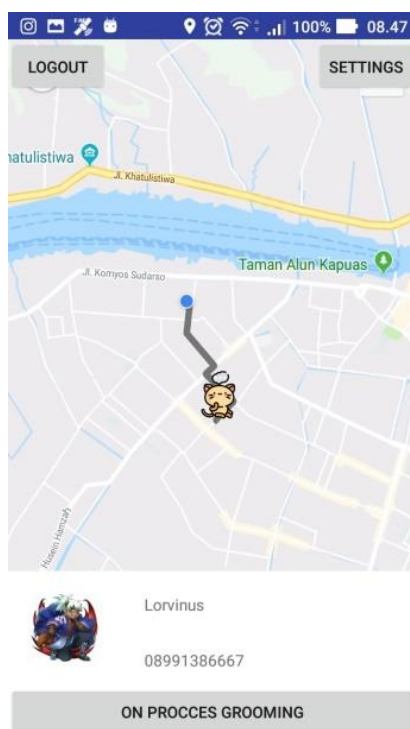
Tahap implementasi meningkatkan kecepatan pembangunan melalui prototyping cepat, utilitas pengguna akhir yang lebih baik, kesederhanaan dan kegunaan dari desain *Graphical User Interface (GUI)*. Pendefinisikan *interface-interface* diantara komponen sistem dan spesifikasi *interface* tidak boleh ambigu. Merancang antarmuka merupakan bagian yang paling penting dari merancang sistem. Biasanya hal tersebut juga merupakan bagian yang paling sulit karena dalam merancang antarmuka harus memenuhi tiga persyaratan : sebuah antarmuka harus sederhana, sebuah antarmuka harus lengkap, dan sebuah antarmuka harus memiliki kinerja yang cepat. Alasan utama mengapa antarmuka sulit untuk dirancang adalah karena setiap antarmuka adalah sebuah bahasa pemrograman yang kecil : antarmuka menjelaskan sekumpulan objek-objek dan operasi-operasi yang bisa digunakan untuk manipulasi objek.



Gambar 6. Tampilan Awal Aplikasi Pemesanan Jasa Grooming Kucing



Gambar 7. Merupakan halaman user jika mendapatkan groomer setelah memesan dgn memilih menu call groomer dan akan menampilkan lokasi groomer.



Gambar 8. Tampilan Ketika Groomer Mendapatkan Pesanan Grooming dan akan menunjukkan arah yg akan dilalui oleh groomer dengan algoritma greedy dan akan menampilkan lokasi user.

4. KESIMPULAN

Berdasarkan hasil kajian dan pembahasan dalam penelitian ini, maka dapat diambil beberapa kesimpulan sebagai berikut, Aplikasi pemesanan jasa grooming kucing ini di buat menggunakan android studio , dengan bahasa pemrograman java , dan menggunakan database firebase. Algoritma greedy digunakan untuk menentukan jalur terpendek yang akan dilalui groomer untuk menuju ke user yang memesan jasa grooming. Permasalahannya yaitu pemilik kucing lebih memilih menggunakan jasa *grooming* daripada melakukan *grooming* sendiri dikarenakan berbagai macam alasan seperti kurangnya pengetahuan tentang *grooming*, tidak adanya waktu untuk *grooming*, dan tidak bisa melakukan *grooming* kucingnya sendiri. Hasil akhir adalah sebuah aplikasi pemesanan jasa grooming kucing untuk area pontianak yang dapat digunakan oleh pemilik kucing yang membutuhkan jasa grooming kucing tanpa membawa kucing mereka ke petshop atau cathouse. Informasi yang tersaji di aplikasi mulai dari informasi mengenai pengguna dan groomer , serta rute yang dilalui beserta tarif ongkir.

5. SARAN

Ada beberapa saran yang dapat penulis berikan berkaitan dengan pengembangan aplikasi, saran yang dapat diberikan adalah koordinat user maupun groomer harus benar-benar valid agar bisa ditampilkan di map secara online. Perlu melakukan update google map API yang merupakan komponen utama untuk menampilkan peta secara online. Jaringan pada android harus bagus untuk mengakses aplikasi pemesanan jasa grooming kucing ini agar *Global Positioning System (GPS)* membaca dengan baik.

DAFTAR PUSTAKA

- [1] Hayati,E,N., Yohanes,A., 2014, Pencarian Rute Terpendek Menggunakan Algoritma Greedy. *Jurnal : Seminar Nasional IENACO*, hal : 391-397 ISSN: 2337-4349, 2014.
- [2] Akhmad,E.G., Setiawan,D.B.,Furqon.T.,2017,Aplikasi Perencanaan Wisata di Malang Raya dengan Algoritma Greedy. *Jurnal : Pengembangan Teknologi Informasi dan Ilmu Komputer*, Vol. 1, No.12, hal : 1459-1467, e-ISSN : 2548-964X, 2017.
- [3] Purnomo,A,C., Yuliana,M., Prasetyaningrum,I., 2012, Implementasi Algoritma Greedy Pada Taksi Wisata Berbasis Web. *Jurnal : Seminar Nasional Komputer dan Sistem Intelijen*, 2014.
- [4] Hartanto,F., Safitri,Y., 2014, Rancang Bangun Aplikasi Pencarian Lokasi Wisata Kota Bogor Menggunakan Algoritma Greedy Berbasis Android. *Jurnal : Techno Nusa Mandiri*, Vol. XI No.2, September 2014.
- [5] Alamsyah,F,B., Ibrahim,M,M., Wigati,P., 2012, Penerapan Algoritma Greedy Studi Kasus Pencarian Rumah Sakit Terdekat Di Jakarta Selatan. *Jurnal : Prosiding Seminar Ilmiah Nasional Komputer dan Sistem Intelijen*, Vol. 7, hal : 559-560, ISSN : 2302-3740, 2012.
- [6] Sugiyono, 2011, *Metode Penelitian Kuantitatif, Kualitatif, dan R&D*. Bandung : Alfabeta.
- [7] Kendal, K.E dan Kendal, J.E, 2010, *Analisis Perancangan Sistem*, PT Index. Klaten.
- [8] Levitin, A., 2010, *Pengantar Desain dan Analisis Algoritma*, Salemba Infotek, Jakarta.
- [9] Munir R., 2005, *Matematika Diskrit. Revisi Kelima*, Informatika, Bandung.
- [10] Munawar, 2005, *Pemodelan Visual dengan UML*, Graha Ilmu, Yogyakarta.
- [11] Munir, Rinaldi, dan Lidya, Leony., 2016, *Algoritma dan Pemrograman Dalam Bahasa Pascal,C,dan C++* (Buku Edisi Keenam), Informatika, Bandung.
- [12] Safaat, 2012, *Android Pemrograman Aplikasi Mobile Smartphone dan Tablet Pc Berbasis Android*, Informatika, Bandung.
- [13] Ristono A. dan Puryani, 2011, *Penelitian Operasional lanjut.Edisi Pertama*, Graha Ilmu, Yogyakarta
- [14] Rachmawati, D dan Chandra, A. 2013. Implementasi Algoritma Greedy Untuk Menyelesaikan Masalah Knapsack Problem. *Jurnal : Saindikom*, Vol. 12, No. 3, September 2013.