

Aplikasi Pencarian Rute Optimal Jasa Pengiriman Barang Menggunakan Metode Ant Colony Optimization (Studi Kasus: TIKI Kubu Raya)

Tedy Rismawan¹⁾, Muhamad Reksy Mulia²⁾, Rahmi Hidayati³⁾

^{1,2,3)}Jurusan Rekayasa Sistem Komputer, Fakultas MIPA, Universitas Tanjungpura
Jl. Prof. Dr. H. Hadari Nawawi, Pontianak

e-mail: tedyrismawan@siskom.untan.ac.id, reksy@untan.ac.id, rahmihidayati@siskom.untan.ac.id

Abstrak

Semakin tingginya minat masyarakat untuk berbelanja secara online menyebabkan perkembangan perusahaan jasa pengiriman barang semakin pesat. Hal ini berdampak pada peningkatan persaingan antar perusahaan jasa pengiriman barang. Untuk mendapatkan kepercayaan konsumen agar tetap digunakan dalam urusan pengantaran barang, pihak perusahaan pengantaran barang harus dapat menjaga ketepatan waktu dalam pengantaran barang. Penelitian ini ditujukan untuk membantu pihak perusahaan pengiriman barang meminimalisir keterlambatan waktu penerimaan barang oleh konsumen dengan membangun aplikasi yang dapat membantu menentukan rute optimal dalam mengantarkan barang. Aplikasi dibangun pada perangkat android dengan mengimplementasikan metode Ant Colony Optimization untuk mendapatkan rute yang optimal. Aplikasi memberikan informasi rute optimal dalam bentuk urutan pengantaran barang yang memiliki jarak tempuh yang paling optimal. Berdasarkan pengujian yang telah dilakukan pada 30 lokasi pengantaran barang, aplikasi menghasilkan urutan rute optimal pengantaran barang dengan total jarak 21.1 km.

Kata kunci: pengiriman barang, ant colony optimization, rute optimal, android

Abstract

The increasing of the public interest to do online shopping causing the rapid development of freight forwarding companies. It has an impact on increasing competition among freight forwarding companies. To get trust from the consumer to always be used in the case of delivering the item, the freight forwarding company must be able to maintain the timelines in the delivering item. This research is intended to help the freight forwarding company minimize the time delay in receiving items by consumers by building applications that can help decide the optimal route for delivering items. The application is built on an android device by implementing the Ant Colony Optimization to get the optimal route. The application provides optimal route information in the form of sequences of locations for delivery of items that have the optimal distance. Based on testing that has been done on 30 locations of delivery items, the application generates the order of delivery route with a total distance of 21.1 km.

Keywords: freight forwarding, ant colony optimization, optimal route, android

1. Pendahuluan

Perusahaan jasa pengiriman barang pada saat ini berkembang dengan cukup cepat, hal ini dapat dilihat dari survei Logistics Performance Index (LPI) dimana Indonesia berada di peringkat 46 dunia pada tahun 2018. Peringkat ini naik cukup signifikan, karena pada tahun 2016 Indonesia masih di peringkat 63 dunia [1]. Meningkatnya nilai LPI tidak lepas dari semakin tingginya minat masyarakat untuk berbelanja secara *online* di e-commerce yang akan berkaitan erat pada proses pengiriman barang. Pengiriman barang saat ini menjadi salah satu bidang bisnis yang cukup menjanjikan, hal ini dapat dilihat dari semakin bertambahnya jumlah

perusahaan yang bergerak di bidang tersebut, yang berarti konsumen akan semakin diuntungkan karena dapat memilih yang terbaik dari perusahaan-perusahaan yang ada. Keadaan ini akan mengakibatkan persaingan yang ketat antar perusahaan dalam memperoleh konsumen, sehingga perusahaan harus berusaha sebaik mungkin untuk melayani konsumen yang menggunakan jasa mereka. Salah satu kendala perusahaan dalam proses pengantaran barang adalah seringkali ditemui para kurir pengantar barang tidak optimal dalam memilih jalur pengantaran barang. Permasalahan ini muncul karena kurir mengantarkan barang hanya berdasarkan pengetahuan serta pengalaman pribadi, dan ini akan semakin memiliki dampak yang besar ketika jumlah penerima barang semakin banyak dengan alamat yang semakin bervariasi. Dampak yang dirasakan langsung apabila menggunakan rute yang tidak optimal adalah proses pengantaran akan semakin lama serta biaya yang digunakan juga semakin besar.

Berdasarkan permasalahan tersebut, maka diperlukan sebuah solusi untuk meminimalisir keterlambatan waktu pengantaran barang, yaitu dengan sebuah aplikasi pencarian rute untuk mendapatkan rute optimal berupa urutan lokasi pengantaran barang yang memiliki jarak paling optimal dalam mencapai seluruh lokasi pengantaran. Biasanya jalur optimal diperoleh dengan menghitung waktu tempuh, ataupun berdasarkan jarak satu lokasi ke lokasi yang lainnya. Semakin banyak lokasi yang akan dikunjungi maka akan semakin rumit menghitung jalur yang optimal [2]. Terkait dengan algoritma yang dapat digunakan untuk mencari rute optimal, Ant Colony Optimization (ACO) merupakan salah satu algoritma yang dapat digunakan karena beberapa kelebihanannya. Koloni semut merupakan algoritma yang bersifat heuristik untuk menyelesaikan permasalahan optimasi. Algoritma ini diinspirasi oleh lingkungan koloni semut pada saat mencari makanan. Semut dapat mencari lintasan terpendek dari suatu sumber makanan menuju sarangnya tanpa harus melihatnya secara langsung. Semut-semut mempunyai penyelesaian yang sangat unik dan sangat maju, yaitu dengan menggunakan jejak pheromon pada suatu jalur untuk berkomunikasi dan membangun solusi, semakin banyak jejak pheromon ditinggalkan maka jalur tersebut akan diikuti oleh semut lain. Pheromone merupakan alat komunikasi berupa hormon yang dikeluarkan oleh semut sebagai penunjuk jalan bagi semut yang lain. Dengan adanya informasi pheromone, maka semut-semut selanjutnya tidak akan berjalan secara acak lagi, namun akan lebih tertarik mengikuti jalur yang ada pheromonenya. Semakin banyak semut melalui suatu jalur, semakin banyak pula jumlah pheromone yang tertinggal di jalur tersebut. Sehingga, lama kelamaan semua semut melalui satu jalur yang seragam, yaitu jalur yang terpendek [3].

Algoritma koloni semut telah digunakan pada penelitian untuk mencari lokasi tempat ibadah terdekat di kota Bandung [4]. Pada penelitian ini dibangun sebuah sistem yang memberikan informasi serta petunjuk arah tempat ibadah di Kota Bandung. Aplikasi yang dibangun terbukti dapat membantu para pengguna khususnya wisatawan dalam mencari tempat ibadah terdekat sehingga dapat mengefisienkan waktu wisatawan yang ingin beribadah ketika sedang melakukan wisata di sekitar Kota Bandung. Algoritma ini juga telah diterapkan pada penelitian untuk memilih objek wisata berbasis android [5]. Pada penelitian ini, algoritma koloni semut digunakan untuk menentukan objek wisata yang paling optimal untuk dikunjungi dengan memperhitungkan variabel biaya pada jarak. Hasil penelitian ini berhasil membuat sistem yang memberi rute optimal berdasarkan biaya perjalanan termurah dalam memilih objek wisata di daerah Kota Karawang. Selain menggunakan algoritma koloni semut, penelitian tentang pencarian rute optimal juga pernah dilakukan menggunakan algoritma djikstra. Pada penelitian ini rute optimal yang dicari adalah rute pengiriman barang pada PT. Kencana Link Nusantara Medan [6]. Penelitian ini menghasilkan sebuah sistem yang mengaplikasikan algoritma djikstra untuk mencari rute terpendek sehingga dapat membantu mengefisienkan proses pengiriman barang pada PT. Kencana Link Nusantara Medan.

Berdasarkan penelitian-penelitian yang telah dilakukan sebelumnya, algoritma koloni semut merupakan salah satu algoritma optimasi yang dapat digunakan untuk membantu menyelesaikan permasalahan pencarian rute optimal. Pada penelitian ini dibuat aplikasi pencarian rute optimal pada jasa pengiriman barang menggunakan algoritma koloni semut. Untuk mempermudah penggunaannya, aplikasi ini dibangun berbasis android karena dapat digunakan dimana pun dan kapan pun oleh penggunanya.

Aplikasi Pencarian Rute Optimal jasa Pengiriman Barang Menggunakan Metode Ant Colony Optimization (Studi Kasus :TIKI Kubu Raya)(Tedy Rismawan, dkk)

2. Metode Penelitian

Metode yang digunakan pada penelitian ini mencakup obeservasi dan pengumpulan data, studi literatur, perancangan sistem, implementasi serta pengujian yang menggunakan metode User Acceptance Test (UAT).

2.1 Observasi dan Pengumpulan Data

Observasi dilakukan untuk mendapatkan data dan informasi yang ada di lapangan. Pada tahapan ini dilakukan aktivitas wawancara terhadap beberapa kurir serta manager dari perusahaan ekspedisi TIKI Kubu Raya. Adapun data yang diperoleh dari proses observasi ini adalah data pengantaran barang yang nantinya akan menjadi data yang akan digunakan pada pengujian sistem. Data pengantaran barang yang digunakan pada penelitian ini sebanyak 30 lokasi.

2.2 Studi Literatur

Studi literatur dilakukan untuk memperoleh informasi yang relevan dengan permasalahan yang akan diselesaikan. Adapun sumber pustaka yang digunakan adalah buku, jurnal penelitian yang memiliki kaitan dengan algoritma koloni semut ataupun proses optimasi jalur pengantaran, serta sumber-sumber lain yang berkaitan dengan penelitian ini.

2.3 Perancangan Sistem

Tahapan ini diperlukan agar sistem yang akan dibangun sesuai dengan yang kebutuhan yang ada, selain itu hasil perancangan dapat dijadikan panduan dalam membangun aplikasi. Proses perancangan pada penelitian dilakukan dengan memodelkan sistem sesuai dengan pemodelan perangkat lunak berorientasi pada objek atau disebut dengan OOP (Object Oriented Programming). Pemodelan pada penelitian ini menggunakan UML (Unified Modelling Language) sebagai Bahasa pemodelannya. Sebuah permodelan bahasa adalah suatu Bahasa dimana kata-kata dan aturannya berfokus pada penggambaran sistem secara konseptual dan fisik [7]. Tujuan dari pemodelan sistem ini adalah untuk mempermudah pengembang dalam melakukan proses pengkodean aplikasi yang dikembangkan [8]. Proses dimulai dengan merancang user interface, use case diagram, class diagram, sequence diagram, dan activity diagram. Aplikasi ini bekerja dengan admin memberikan masukan (*input*) lokasi-lokasi pengantaran barang dan kemudian akan masuk ke dalam device kurir yang selanjutnya akan diproses menggunakan metode Ant Colony Optimization untuk mendapatkan rute optimal berupa urutan lokasi pengantaran dalam mengantarkan barang ke seluruh konsumen dalam satu kali perjalanan oleh kurir.

2.4 Implementasi

Pada tahapan ini, segala hal yang telah dirancang pada bagian perancangan sistem akan diimplementasikan, sehingga dapat menghasilkan sistem yang sesuai dengan kebutuhan. Langkah awal pada tahapan ini adalah melakukan integrasi agar sistem dengan Application Programming Interface (API) Google Firebase dan Google Maps dapat terhubung. Hal ini dilakukan agar sistem dapat menampilkan lokasi pengantaran pada Google Maps berdasarkan database yang ada pada Google Firebase. Selanjutnya, engine utama dari aplikasi akan dibangun menggunakan perangkat lunak Android Studio yang hasilnya akan memberikan rute optimal berupa urutan lokasi pengantaran barang oleh kurir kepada seluruh konsumen dalam satu kali perjalanan.

2.5 Pengujian Sistem

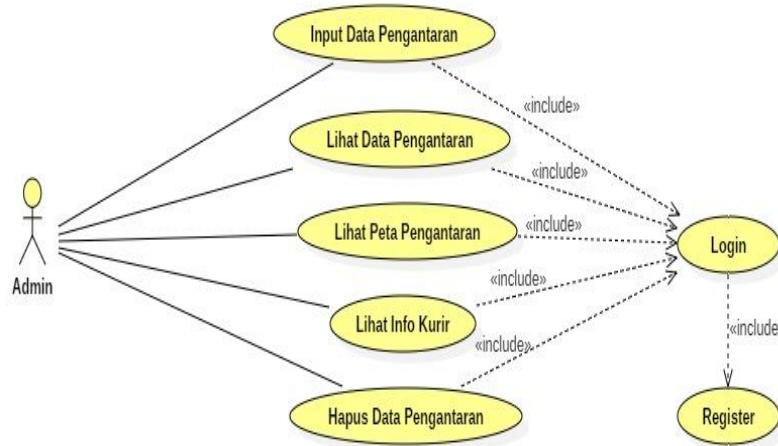
Tahapan pengujian dilakukan untuk memastikan apakah sistem yang telah dibangun berfungsi dengan baik dan menghasilkan keluaran sesuai dengan yang diharapkan. Pengujian dilakukan dengan cara memasukkan data yang telah didapat ke dalam aplikasi dan membandingkan hasil yang diperoleh dari aplikasi dengan hasil yang perhitungan yang dilakukan secara manual. Selanjutnya dilakukan juga pengujian User Acceptance Testing (UAT) untuk mengevaluasi sejauh mana kesiapan sistem sebelum digunakan oleh pengguna. Kriteria pengujian pada UAT merupakan kriteria yang wajib disediakan oleh sistem agar dapat diterima oleh pengguna [9], [10].

3. Perancangan

Sistem yang dibangun pada penelitian ini dirancang menggunakan Unified Modelling Language (UML) dimana di dalamnya terdapat beberapa diagram untuk menggambarkan aksi dari entitas yang ada di dalam sistem serta interaksi yang ada di dalamnya.

3.1. Use Case Diagram

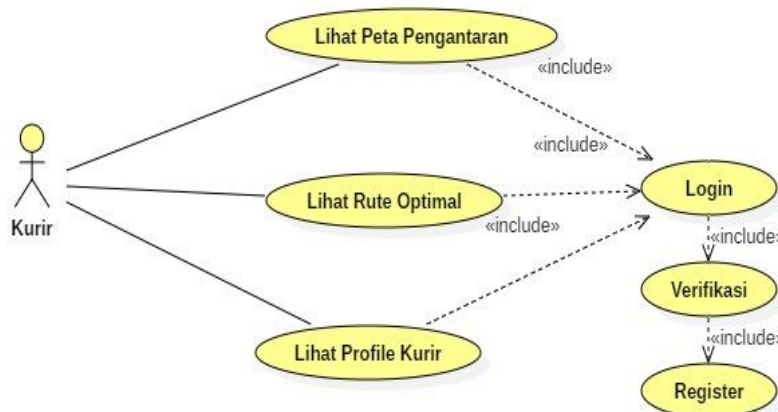
Use case diagram digunakan untuk menggambarkan fungsionalitas dari sistem yang akan dibangun. Pada sistem yang akan dibangun ini terdapat dua aktor yaitu admin dan kurir. Use case diagram admin dapat dilihat pada Gambar 1.



Gambar 1 Use case diagram admin

Untuk mulai menggunakan aplikasi, maka admin harus melakukan proses registrasi terlebih dahulu agar akun admin direkam di dalam sistem. Setelah proses registrasi berhasil dilakukan, maka admin dapat melakukan berbagai aktivitas dengan login terlebih dahulu ke dalam sistem berdasarkan data yang telah dimasukkan saat proses register. Adapun aktivitas yang dapat dilakukan oleh admin pada sistem adalah *input* data pengantaran, lihat data pengantaran, lihat peta pengantaran, lihat informasi kurir dan hapus data pengantaran

Selain user admin, aktor lain di dalam sistem ini adalah kurir. Gambar 2 merupakan use case diagram dari aktor kurir.



Gambar 2 Use case diagram kurir

Adapun aktivitas yang dapat dilakukan oleh kurir adalah:

1. Lihat peta pengantaran

Pada bagian ini kurir dapat melihat peta yang berisi lokasi-lokasi pengantaran barang kurir tersebut. Data ini diperoleh dari data pengantaran yang telah dimasukkan oleh admin.

2. Lihat rute optimal

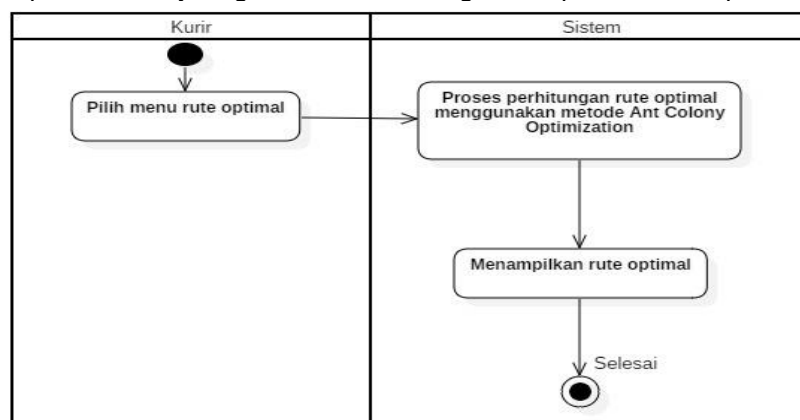
Di bagian ini kurir dapat menampilkan rute optimal serta data pengantaran barang kurir. Rute optimal yang tampil merupakan urutan lokasi yang harus dikunjungi oleh kurir yang mana nilainya merupakan hasil perhitungan menggunakan algoritma koloni semut.

3. Lihat profil

Bagian ini merupakan tempat dimana kurir dapat melihat informasi terkait dengan akun yang sedang digunakan. Adapun informasi ini merupakan informasi yang dimasukkan oleh kurir pada saat proses registrasi.

3.2. Activity Diagram

Pada activity diagram digambarkan aliran kerja atau aktivitas dari sistem. Selain itu, activity diagram juga menggambarkan proses logika dan proses bisnis pada suatu sistem. Gambar 3 merupakan activity diagram ketika kurir ingin memperoleh rute optimal.



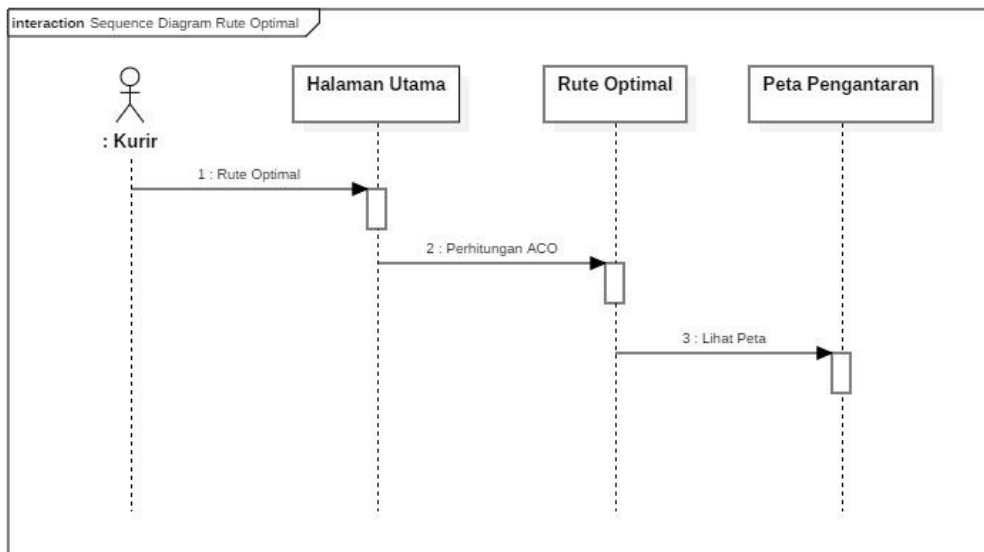
Gambar 3 Activity diagram rute optimal

Adapun aliran aktivitas yang terjadi adalah:

1. Kurir memilih menu rute optimal untuk melihat rute optimal berupa urutan lokasi pengantaran yang memiliki jarak paling optimal
2. Sistem akan melakukan perhitungan rute optimal yang akan dituju oleh kurir dengan menggunakan algoritma koloni semut.
3. Setelah sistem selesai menghitung rute optimal didalam aplikasi, maka aplikasi akan menampilkan rute optimal berupa urutan lokasi pengantaran barang yang memiliki jarak paling optimal berdasarkan perhitungan algoritma koloni semut.

3.3. Sequence Diagram

Pada sequence diagram digambarkan interaksi antar objek di dalam dan di sekitar sistem berupa pesan yang digambarkan terhadap waktu. Gambar 4 merupakan sequence diagram dari proses untuk memperoleh rute optimal.

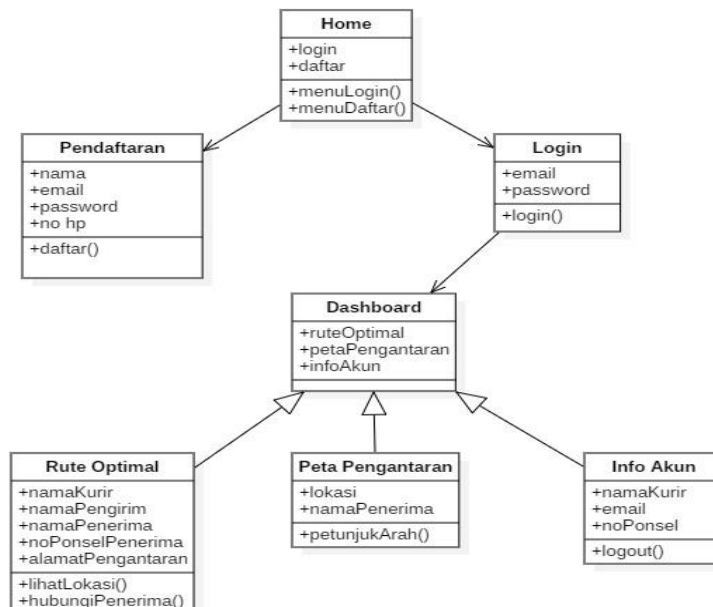


Gambar 4 Sequence diagram memperoleh rute optimal

Proses lihat rute optimal dilakukan pada aplikasi yang berjalan pada aktor kurir. Rute optimal merupakan rute pengantaran barang yang didapat dari perhitungan jarak pada data pengantaran barang dengan menggunakan algoritma koloni semut yang memiliki jarak paling optimal dalam mengantarkan seluruh barang yang dibawa oleh kurir. Pada proses ini juga dihitung total jarak yang dibutuhkan untuk mengantar seluruh barang yang dibawa oleh kurir.

3.4. Class Diagram

Class diagram digunakan untuk menggambarkan class-class objek yang menyusun sistem serta hubungan antara class dengan objek lainnya. Class diagram pada penelitian ini dibagi menjadi 2, yaitu class diagram untuk admin dan class diagram untuk kurir.



Gambar 5 Class Diagram

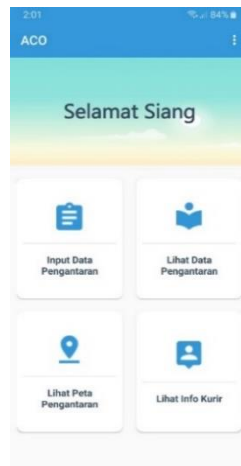
Pada gambar 5, class diagram yang disajikan merupakan class diagram untuk aktor kurir dimana terdapat beberapa kelas yaitu home, pendaftaran, login, dashboard, rute optimal, peta pengantaran serta info akun. Pada tiap kelas memiliki atribut dan method yang digunakan pada proses di sistem

4. Implementasi Sistem

Tahap implementasi merupakan proses menerjemahkan proses perancangan ke dalam kode program sehingga aplikasi dapat digunakan. Adapun Bahasa pemrograman yang digunakan untuk membangun sistem ini adalah Java dengan database Firebase.

4.1 Antarmuka Sistem

Pengguna dapat menggunakan sistem dengan cara berinteraksi dengan antarmuka sistem. Secara umum antarmuka sistem dibagi menjadi 2, yaitu untuk admin dan untuk kurir. Namun untuk mulai mengakses sistem, kedua entitas tersebut harus melakukan proses registrasi dan login terlebih dahulu. Setelah berhasil melakukan login, maka tampilan dari sistem akan menyesuaikan dengan otoritas pengguna. Apabila pengguna login sebagai admin maka akan muncul *dashboard* admin (gambar 6) dimana di dalamnya terdapat menu yang dapat dipilih oleh admin, yaitu *input* data pengantaran, lihat data pengantaran (gambar 7), lihat peta pengantaran dan lihat info kurir.

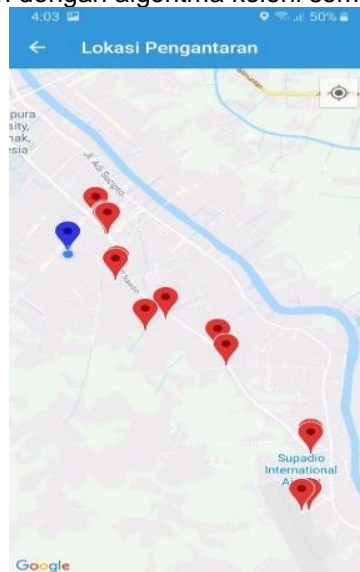


Gambar 6 Dashboard Admin



Gambar 7 Data Pengantaran

Sedangkan apabila pengguna masuk ke dalam sistem sebagai kurir, maka akan muncul *dashboard* kurir yang mana di dalamnya terdapat menu lihat peta pengantaran (gambar 8), rute optimal hasil perhitungan dengan algoritma koloni semut (gambar 9) dan profile.



Gambar 8 Peta Pengantaran Barang



Gambar 9 Rute Optimal hasil ACO

Pada gambar 9 terlihat urutan alamat dari penerima paket kiriman yang merupakan urutan yang paling optimal berdasarkan perhitungan algoritma koloni semut/ACO. Berdasarkan hasil perhitungan ini, kurir dapat mengantarkan paket kiriman dengan lebih efisien baik dari sisi waktu maupun biaya.

4.2 Hasil Perhitungan Algoritma Koloni Semut

Perhitungan dilakukan dengan menggunakan data pengantaran barang oleh kurir dari jasa pengiriman barang TIKI. Data yang telah diperoleh akan diambil koordinat lokasinya menggunakan google maps untuk mendapatkan *latitude* dan *longitude* dari lokasi pengantaran

barang tersebut. Setelah *latitude* dan *longitude* dari lokasi tersebut diperoleh, selanjutnya diproses menggunakan algoritma koloni semut.

Tabel 1 Data alamat penerima kiriman

No	Penerima	Latitude	Longitude	No	Penerima	Latitude	Longitude
1	PNS Nurhayati	-0.1465272	109.405799	16	Kopda Supandi	-0.1465272	109.405799
2	Yohanes Eko K	-0.1465272	109.405799	17	Linda	-0.0869986	109.36710219
3	Abdul Malik	-0.1467338	109.4043446	18	Wahana Inti Niaga	-0.075038	109.36513
4	Ratna Pertiwi	-0.0709836	109.36315049	19	Tafa	-0.0863558	109.3673881
5	Gudang Astra	-0.10525509	109.3875386	20	Saiful	-0.099982	109.37306389
6	Wilson Ginting	-0.0709836	109.36315049	21	Saleh Santoso	-0.086044	109.37252799
7	Fany Alfinda	-0.1320171	109.406139	22	Aryo Pratama	-0.0686839	109.37177129
8	Edi Santoso	-0.1091896	109.3892065	23	Ringo Renaldi	-0.07363839	109.3666025
9	Marcella	-0.0709836	109.36315049	24	Iwan	-0.0931617	109.380656
10	Serka Triyatno	-0.1465272	109.405799	25	Rustam Efendi	-0.09893049	109.3852646
11	Agnes Bp Atmo	-0.0970220	109.3771319	26	Burhanuddin	-0.10454569	109.3916218
12	Ibu Susi	-0.13114099	109.406165	27	Raka Fajrianto	-0.1113854	109.3980914
13	Agustino S	-0.0746150	109.364977	28	Rian Dinata	-0.07341439	109.3726492
14	RND	-0.07504060	109.36554810	29	Arif Pambudi	-0.12094129	109.4008020
15	Tata	-0.0863558	109.3673881	30	Alvin Hendra	-0.1349321	109.4093103

Tabel 1 merupakan data penerima kiriman berdasarkan koordinat yang diambil dari alamat dan disesuaikan dengan google maps. Setelah mendapatkan koordinat alamat dari penerima maka tahapan yang dilakukan untuk mendapatkan rute pengantaran yang optimal adalah sebagai berikut:

1. Hitung jarak antara satu lokasi dengan lokasi lainnya serta jarak tiap lokasi pengantaran dengan gudang dimana barang akan diberangkatkan. Proses perhitungan jarak ini menggunakan persamaan Euclidian. Contoh hasil perhitungan jarak dapat dilihat pada tabel 2.

Tabel 2 Contoh hasil perhitungan jarak

Lokasi	Gudang	1	2	3	4	5
Gudang	0	14.78	14.78	14.72	5.13	9.77
1	14.78	0	0	0.16	9.66	5.02
2	14.78	0	0	0.16	9.66	5.02
3	14.72	0.16	0.16	0	9.60	4.98
4	5.13	9.66	9.66	9.60	0	4.68
5	9.77	5.02	5.02	4.98	4.68	0

2. Tahapan selanjutnya adalah menghitung nilai visibilitas antar lokasi untuk digunakan pada perhitungan probabilitas antar lokasi satu dengan lokasi lainnya. Jika jarak antar

lokasi = 0, maka $\eta_{ij} = 0$. Nilai visibilitas antar lokasi didapat dari 1 dibagi dengan jarak antar lokasi. Tabel 3 menyajikan contoh perhitungan nilai visibilitas.

Tabel 3 Contoh hasil perhitungan nilai visibilitas

η_{ij}	Gudang	1	2	3	4	5
Gudang	0	0.07	0.07	0.07	0.19	0.10
1	0.07	0	0	6.12	0.10	0.20
2	0.07	0	0	6.12	0.10	0.20
3	0.07	6.12	6.12	0	0.10	0.20
4	0.19	0.10	0.10	0.10	0	0.20
5	0.10	0.20	0.20	0.20	0.21	0

3. Setelah mendapatkan nilai visibilitas, tahapan selanjutnya adalah menghitung nilai probabilitas antar satu lokasi dengan lokasi lainnya untuk kemudian nilai probabilitas tertinggi akan dimasukkan ke dalam *tabu list* dari masing-masing semut. Perhitungan nilai probabilitas antar lokasi satu dengan lokasi lainnya dilakukan dengan menggunakan nilai dari parameter-parameter yang telah ditentukan dan juga nilai dari visibilitas antar lokasi satu dengan lokasi lainnya yang telah dihitung sebelumnya. Perhitungan ini dilakukan dengan menggunakan persamaan:

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum [\tau_{ij'}]^\alpha \cdot [\eta_{ij'}]^\beta} \text{ untuk } j \in \{N - \text{Tabu}_k\}$$

Contoh hasil dari perhitungan probabilitas antar lokasi dapat dilihat pada tabel 4

Tabel 4 Contoh hasil perhitungan probabilitas

Probabilitas	Gudang	1	2	3	4	5
Gudang	0	0.000676631	0.000676631	0.0006791	0.001947974	0.00102322
1	0.000676631	0	0	0.061151725	0.001035515	0.00199046
2	0.000676631	0	0	0.061151725	0.001035515	0.00199046
3	0.0006791	0.061151725	0.061151725	0	0.001041811	0.002007236
4	0.001947974	0.001035515	0.001035515	0.001041811	0	0.00213564
5	0.00102322	0.00199046	0.00199046	0.002007236	0.00213564	0

4. Kemudian tahapan terakhir adalah memasukkan lokasi-lokasi yang telah dituju oleh semut kedalam *tabu list* untuk tiap semut. Setelah itu hitung total jarak yang dilalui oleh semut untuk mencapai seluruh lokasi dan dibandingkan dengan total jarak yang telah ditempuh oleh masing-masing semut. Semut yang melalui seluruh lokasi dengan total jarak paling pendek merupakan solusi dari permasalahan. Adapun total jarak yang ditempuh dapat dilihat pada tabel 5.

Tabel 5 Perhitungan total jarak perjalanan

Semut Ke	Rute	Total Jarak(km)
1	gudang - 1 - 2 - 10 - 16 - 3 - 30 - 7 - 12 - 29 - 27 - 8 - 5 - 26 - 25 - 24 - 11 - 20 - 21 - 15 - 17 - 19 - 14 - 18 - 13 - 23 - 4 - 6 - 9 - 22 - 28	30.1
2	gudang - 2 - 1 - 10 - 16 - 3 - 30 - 7 - 12 - 29 - 27 - 8 - 5 - 26 - 25 - 24 - 11 - 20 - 21 - 15 - 17 - 19 - 14 - 18 - 13 - 23 - 4 - 6 - 9 - 22 - 28	30.1
3	gudang - 3 - 1 - 10 - 16 - 3 - 30 - 7 - 12 - 29 - 27 - 8 - 5 - 26 - 25 - 24 - 11 - 20 - 21 - 15 - 17 - 19 - 14 - 18 - 13 - 23 - 4 - 6 - 9 - 22 - 28	29.9
4	gudang - 4 - 6 - 9 - 13 - 18 - 14 - 23 - 28 - 22 - 21 - 15 - 19 - 17 - 11 - 20 - 24 - 25 - 5 - 26 - 8 - 27 - 29 - 12 - 7 - 30 - 1 - 2 - 10 - 16 - 3	21.1
5	gudang - 5 - 26 - 8 - 27 - 29 - 12 - 7 - 30 - 1 - 2 - 16 - 10 - 3 - 25 - 24 - 11 - 20 - 21 - 15 - 17 - 19 - 14 - 18 - 13 - 23 - 4 - 6 - 9 - 22 - 28	29.9

6	gudang - 6 - 4 - 9 - 13 - 18 - 14 - 23 - 28 - 22 - 21 - 15 - 19 - 17 - 11 - 20 - 24 - 25 - 5 - 26 - 8 - 27 - 29 - 12 - 7 - 30 - 1 - 2 - 10 - 16 - 3	21.1
7	gudang - 7 - 12 - 30 - 1 - 2 - 10 - 16 - 3 - 29 - 27 - 8 - 5 - 26 - 25 - 24 - 11 - 20 - 21 - 15 - 17 - 19 - 14 - 18 - 13 - 23 - 4 - 6 - 9 - 22 - 28	30.3
8	gudang - 8 - 5 - 26 - 25 - 24 - 11 - 20 - 21 - 15 - 19 - 17 - 14 - 18 - 13 - 23 - 4 - 6 - 9 - 22 - 28 - 27 - 29 - 12 - 7 - 30 - 1 - 2 - 10 - 16 - 3	29.5
9	gudang - 9 - 4 - 6 - 13 - 18 - 14 - 23 - 28 - 22 - 21 - 15 - 19 - 17 - 11 - 20 - 24 - 25 - 5 - 26 - 8 - 27 - 29 - 12 - 7 - 30 - 1 - 2 - 10 - 16 - 3	21.1
10	gudang - 10 - 1 - 2 - 16 - 3 - 30 - 7 - 12 - 29 - 27 - 8 - 5 - 26 - 25 - 24 - 11 - 20 - 21 - 15 - 17 - 19 - 14 - 18 - 13 - 23 - 4 - 6 - 9 - 22 - 28	30.1
11	gudang - 11 - 20 - 24 - 25 - 5 - 26 - 8 - 27 - 29 - 12 - 7 - 30 - 1 - 2 - 10 - 16 - 3 - 21 - 15 - 17 - 19 - 14 - 18 - 13 - 23 - 4 - 6 - 9 - 22 - 28	30.2
12	gudang - 12 - 7 - 30 - 1 - 2 - 10 - 16 - 3 - 29 - 27 - 8 - 5 - 26 - 25 - 24 - 11 - 20 - 21 - 15 - 17 - 19 - 14 - 18 - 13 - 23 - 4 - 6 - 9 - 22 - 28	30.1
13	gudang - 13 - 18 - 14 - 23 - 4 - 6 - 9 - 22 - 28 - 21 - 15 - 19 - 17 - 11 - 20 - 24 - 25 - 5 - 26 - 8 - 27 - 29 - 12 - 7 - 30 - 1 - 2 - 10 - 16 - 3	21.3
14	gudang - 14 - 18 - 13 - 23 - 4 - 6 - 9 - 22 - 28 - 21 - 15 - 19 - 17 - 11 - 20 - 24 - 25 - 5 - 26 - 8 - 27 - 29 - 12 - 7 - 30 - 1 - 2 - 10 - 16 - 3	21.3
15	gudang - 15 - 19 - 17 - 21 - 24 - 11 - 20 - 25 - 5 - 26 - 8 - 27 - 29 - 12 - 7 - 30 - 1 - 2 - 10 - 16 - 3 - 28 - 22 - 23 - 14 - 18 - 13 - 4 - 6 - 9	29.5
16	gudang - 16 - 1 - 2 - 10 - 3 - 30 - 7 - 12 - 29 - 27 - 8 - 5 - 26 - 25 - 24 - 11 - 20 - 21 - 15 - 17 - 19 - 14 - 18 - 13 - 23 - 4 - 6 - 9 - 22 - 28	30.1
17	gudang - 17 - 15 - 19 - 21 - 24 - 11 - 20 - 25 - 5 - 26 - 8 - 27 - 29 - 12 - 7 - 30 - 1 - 2 - 10 - 16 - 3 - 28 - 22 - 23 - 14 - 18 - 13 - 4 - 6 - 9	29.6
18	gudang - 18 - 14 - 13 - 23 - 4 - 6 - 9 - 22 - 28 - 21 - 15 - 19 - 17 - 11 - 20 - 24 - 25 - 5 - 26 - 8 - 27 - 29 - 12 - 7 - 30 - 1 - 2 - 10 - 16 - 3	21.4
19	gudang - 19 - 15 - 17 - 21 - 24 - 11 - 20 - 25 - 5 - 26 - 8 - 27 - 29 - 12 - 7 - 30 - 1 - 2 - 10 - 16 - 3 - 28 - 22 - 23 - 14 - 18 - 13 - 4 - 6 - 9	29.5
20	gudang - 20 - 11 - 24 - 25 - 5 - 26 - 8 - 27 - 29 - 12 - 7 - 30 - 1 - 2 - 10 - 16 - 3 - 21 - 15 - 17 - 19 - 14 - 18 - 13 - 23 - 4 - 6 - 9 - 22 - 28	29.7
21	gudang - 21 - 15 - 19 - 17 - 14 - 18 - 13 - 23 - 4 - 6 - 9 - 22 - 28 - 24 - 11 - 20 - 25 - 5 - 26 - 8 - 27 - 29 - 12 - 7 - 30 - 1 - 2 - 10 - 16 - 3	23.6
22	gudang - 22 - 28 - 23 - 14 - 18 - 13 - 4 - 6 - 9 - 15 - 19 - 17 - 21 - 24 - 11 - 20 - 25 - 5 - 26 - 8 - 27 - 29 - 12 - 7 - 30 - 1 - 2 - 10 - 16 - 3	21.8
23	gudang - 23 - 14 - 18 - 13 - 4 - 6 - 9 - 22 - 28 - 21 - 15 - 19 - 17 - 11 - 20 - 24 - 25 - 5 - 26 - 8 - 27 - 29 - 12 - 7 - 30 - 1 - 2 - 10 - 16 - 3	21.3
24	gudang - 24 - 11 - 20 - 25 - 5 - 26 - 8 - 27 - 29 - 12 - 7 - 30 - 1 - 2 - 10 - 16 - 3 - 21 - 15 - 17 - 19 - 14 - 18 - 13 - 23 - 4 - 6 - 9 - 22 - 28	30
25	gudang - 25 - 5 - 26 - 8 - 27 - 29 - 12 - 7 - 30 - 1 - 2 - 10 - 16 - 3 - 20 - 11 - 24 - 21 - 15 - 17 - 19 - 14 - 18 - 13 - 23 - 4 - 6 - 9 - 22 - 28	29.3
26	gudang - 26 - 5 - 8 - 27 - 29 - 12 - 7 - 30 - 1 - 2 - 16 - 10 - 3 - 25 - 24 - 11 - 20 - 21 - 15 - 17 - 19 - 14 - 18 - 13 - 23 - 4 - 6 - 9 - 22 - 28	29.9
27	gudang - 27 - 8 - 5 - 26 - 25 - 24 - 11 - 20 - 21 - 15 - 19 - 17 - 14 - 18 - 13 - 23 - 4 - 6 - 9 - 22 - 28 - 29 - 12 - 7 - 30 - 1 - 2 - 10 - 16 - 3	31.2
28	gudang - 28 - 22 - 23 - 14 - 18 - 13 - 4 - 6 - 9 - 15 - 19 - 17 - 21 - 24 - 11 - 20 - 25 - 5 - 26 - 8 - 27 - 29 - 12 - 7 - 30 - 1 - 2 - 10 - 16 - 3	21.4
29	gudang - 29 - 27 - 8 - 5 - 26 - 25 - 24 - 11 - 20 - 21 - 15 - 19 - 17 - 14 - 18 - 13 - 23 - 4 - 6 - 9 - 22 - 28 - 12 - 7 - 30 - 1 - 2 - 10 - 16 - 3	33.4
30	gudang - 30 - 7 - 12 - 29 - 27 - 8 - 5 - 26 - 25 - 24 - 11 - 20 - 21 - 15 - 19 - 17 - 14 - 18 - 13 - 23 - 4 - 6 - 9 - 22 - 28 - 3 - 1 - 2 - 10 - 16	36.6

Pada tabel 5 dapat dilihat bahwa rute terpendek diperoleh oleh semut ke 4, 6 dan 9 dengan total Panjang rute adalah 21,1 km.

5. Pengujian

5.1 Pengujian Hasil Perhitungan Sistem

Pengujian ini bertujuan untuk membandingkan hasil yang didapat oleh aplikasi dengan hasil yang didapat dengan menggunakan perhitungan manual. Hasil perhitungan yang dilakukan oleh aplikasi akan ditampilkan pada jendela *run Android Studio*. Adapun hasil perhitungan pencarian rute optimal dari semut nomor 4 (tabel 5) dapat dilihat pada gambar 10.

```
I/System.out: Tujuan ke 1 = Lokasi 4
I/System.out: Tujuan ke 2 = Lokasi 6
I/System.out: Tujuan ke 3 = Lokasi 9
I/System.out: Tujuan ke 4 = Lokasi 13
I/System.out: Tujuan ke 5 = Lokasi 18
I/System.out: Tujuan ke 6 = Lokasi 14
I/System.out: Tujuan ke 7 = Lokasi 23
I/System.out: Tujuan ke 8 = Lokasi 28
I/System.out: Tujuan ke 9 = Lokasi 22
I/System.out: Tujuan ke 10 = Lokasi 21
I/System.out: Tujuan ke 11 = Lokasi 15
I/System.out: Tujuan ke 12 = Lokasi 19
I/System.out: Tujuan ke 13 = Lokasi 17
I/System.out: Tujuan ke 14 = Lokasi 11
I/System.out: Tujuan ke 15 = Lokasi 20
I/System.out: Tujuan ke 16 = Lokasi 24
I/System.out: Tujuan ke 17 = Lokasi 25
I/System.out: Tujuan ke 18 = Lokasi 5
I/System.out: Tujuan ke 19 = Lokasi 26
I/System.out: Tujuan ke 20 = Lokasi 8
I/System.out: Tujuan ke 21 = Lokasi 27
I/System.out: Tujuan ke 22 = Lokasi 29
I/System.out: Tujuan ke 23 = Lokasi 12
I/System.out: Tujuan ke 24 = Lokasi 7
I/System.out: Tujuan ke 25 = Lokasi 30
I/System.out: Tujuan ke 26 = Lokasi 1
I/System.out: Tujuan ke 27 = Lokasi 2
I/System.out: Tujuan ke 28 = Lokasi 10
I/System.out: Tujuan ke 29 = Lokasi 16
I/System.out: Tujuan ke 30 = Lokasi 3
I/System.out: Total Jarak = 21.1 Km
```

Gambar 10 Hasil perhitungan sistem

Dari gambar 10 dapat dilihat bahwa sistem menghasilkan keluaran yang sama dengan nilai perhitungan manual yang dilakukan sebelumnya dimana hasilnya dapat dilihat pada tabel 5. Hal ini menunjukkan bahwa sistem yang telah dibangun berhasil menemukan rute optimal sesuai dengan algoritma koloni semut.

5.2 Pengujian User Acceptance Test (UAT)

Pengujian UAT dilakukan sebagai bentuk pengujian sistem oleh pengguna untuk melihat sejauh mana sistem yang dibuat dapat diterima oleh pengguna. UAT juga bertujuan untuk memverifikasi sistem terhadap kebutuhan bisnis yang ada, dalam hal ini pengantaran barang. Proses pengujian UAT dilakukan oleh pengguna yang memang sudah paham dengan proses bisnis dan familir dalam menjalankannya, pada permasalahan ini pengguna adalah admin dan kurir. Adapun hasil pengujian UAT dapat dilihat pada tabel 6.

Tabel 6 Hasil pengujian UAT

No	Deskripsi	Skala Kesesuaian				
		STS	TS	CS	S	SS
1	Pengguna mudah melakukan proses pendaftaran/registrasi				11,5%	88,5%
2	Navigasi sistem dapat langsung digunakan dengan mudah			2,9%	5,7%	91,4%

3	Admin dapat memasukkan dan mengakses data pengantaran dengan mudah				14,3%	85,7%
4	Kurir dapat memperoleh data alamat pengantaran				7,2%	92,8%
5	Kurir dapat mengakses rute optimal pengantaran barang dengan mudah				10,8%	89,2%
6	Perangkat android yang dimiliki tidak menemukan masalah dalam mengakses sistem				17,2%	82,8%
Rata-rata					0,49%	11,11%

Ket:

STS = Sangat Tidak Sesuai

TS = Tidak Sesuai

CS = Cukup Sesuai

S = Sesuai

SS = Sangat Sesuai

6. Kesimpulan

Berdasarkan pembahasan yang telah dilakukan pada sistem pencarian rute optimal dengan algoritma semut untuk proses pengantaran barang, maka dapat diambil beberapa kesimpulan:

1. Algoritma koloni semut dapat digunakan untuk menghasilkan rute optimal pada proses pengantaran barang dimana telah dilakukan ujicoba untuk 30 lokasi pengantaran dan menghasilkan rute terbaik dengan panjang rute 21.1 km.
2. Sistem dapat menampilkan urutan lokasi pengantaran barang berdasarkan rute optimal yang dihitung dengan algoritma koloni semut, dan lokasi pengantaran dapat disajikan dalam bentuk peta yang telah terintegrasi dengan Google Maps.
3. Sistem dapat diterima oleh pengguna dengan baik, hal ini dapat dilihat dari nilai rata-rata User Acceptance Test yang mencapai 88,4% pada kategori nilai Sangat Sesuai.

Referensi

- [1] World Bank, "International LPI," 2018. <https://lpi.worldbank.org/international/global/2018> (accessed Jan. 03, 2020).
- [2] I. G. A. D. Saryanti, "Perancangan Simulasi Optimasi Masalah Transportasi Pengiriman Barang Dengan Menggunakan Algoritma Genetika," *J. Sist. Dan Inform.*, vol. 10, pp. 52–63, 2015.
- [3] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, 1997, doi: 10.1109/4235.585892.
- [4] A. Zarman, M. Irfan, and W. Uriawan, "Implementasi Algoritma Ant Colony Optimization pada Aplikasi Pencarian Lokasi Tempat Ibadah Terdekat di Kota Bandung," *J. Online Inform.*, vol. 1, no. 1, p. 6, 2016, doi: 10.15575/join.v1i1.4.
- [5] E. Nurlaelasari, Supriyadi, and U. T. Lenggana, "Penerapan Algoritma Ant Colony Optimization Menentukan Nilai Optimal Dalam Memilih Objek Wisata Berbasis Android," *Simetris*, vol. 9, no. 1, pp. 287–298, 2018.
- [6] A. Z. Hasibuan, "Penentuan Alur Terpendek Pengiriman Barang PT.Kencana Link Nusantara Medan Dengan Algoritma Dijkstra," *J. Ris. Komput.*, vol. 3, no. 6, pp. 14–19, 2016.
- [7] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide (2nd Edition)*. United State: Addison Wesley Professional, 25AD.
- [8] R. S. Pressman, *Software Engineering: A Practitioner's Approach 8th Edition* No Title. New York: McGraw-Hill Education, 2005.
- [9] H. K. N. Leung and P. W. L. Wong, "A Study of User Acceptance Tests," *Softw. Quallity J.* 6, pp. 137–149, 1997.
- [10] P. Hsia, D. Kung, and C. Sell, "Software Requirements and Acceptance Testing," *Ann. Softw. Eng.* 3, pp. 291–317, 1997.