



Penerapan Algoritma Dijkstra Pada Aplikasi Pencarian Fasilitas Pelayanan Kesehatan Terdekat Kota Depok

¹Arif Budiarto, ²Windu Gata, ³Eni Heni Hermaliani, ⁴Agus Salim, ⁵Cicik Sri Rahayu
^{1, 2, 3, 4, 5} Ilmu Komputer, Komputer, STMIK Nusa Mandiri

Alamat Surat

Email: 14002437@nusamandiri.ac.id

Article History:

Diajukan: 30-03-2021; Direvisi: 14-04-2021; Diterima: 28-04-2021

ABSTRAK

Kesehatan menjadi Salah satu faktor utama dalam masyarakat, dimana jarak yang dekat adalah menjadi pilihan utama masyarakat untuk memeriksakan kesehatannya. Dalam penelitian ini akan mencari rute terpendek dari lokasi user ke fasilitas pelayanan kesehatan tujuan, dimana user akan dapat menghemat waktu dan juga biaya transportasi serta mengurangi antrian. Penggunaan Algoritma Dijkstra pada Aplikasi Pencarian Tempat Fasilitas Pelayanan dipilih untuk diterapkan kedalam aplikasi berbasis android untuk menyelesaikan masalah secara langkah demi langkah dalam pencarian rute, jarak dan fasilitas pelayanan kesehatan terdekat. Pendekatan yang digunakan di dalam algoritma Dijkstra membuat pilihan yang terlihat memberikan perolehan terbaik.

Kata kunci: Kesehatan; android; algoritma dijkstra.

ABSTRACT

Health is one of the main factors in society, where a short distance is the main choice for people to check their health. In this study, it will look for the shortest route from the user's location to the destination health service facility, where the user will be able to save time and transportation costs and reduce queues. The use of the Dijkstra Algorithm in the Service Facility Place Search Application was chosen to be applied into an Android-based application to solve problems step by step in searching for routes, distances and closest health service facilities. The approach used in Dijkstra's algorithm makes the choice that seems to give the best return.

Keywords: Health; android; dijkstra's algorithm.

1. PENDAHULUAN

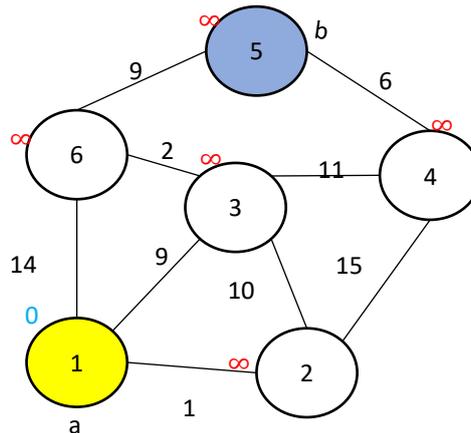
Salah satu faktor yang mempengaruhi rendahnya pemanfaatan pelayanan kesehatan adalah jarak yang ditempuh dan antrian. Dengan semakin canggihnya perkembangan teknologi setiap tahunnya, maka semakin di kembangkannya beragam aplikasi dengan kelebihannya. Aplikasi Pencarian Tempat Fasilitas Pelayanan Kesehatan Terdekat ini menerapkan Algoritma Dijkstra, sehingga salah satu manfaatnya dapat digunakan oleh masyarakat terutama warga Depok untuk memeriksakan kesehatannya di fasilitas pelayanan kesehatan dengan mencari rute terdekat. Algoritma Dijkstra menggunakan prinsip Greedy, mencari jalur terpendek dari satu node (titik/verteks) ke node lain yang searah (directed graph) dimulai dari node asal sampai dengan node yang dituju. Setiap node yang dihitung didapat dari beberapa lokasi strategis yang dapat dikenal seperti lokasi umum yang mudah diingat (Sunardi, 2019).

Berdasarkan uraian diatas maka penelitian ini akan membahas penerapan algoritma dijkstra ke dalam aplikasi Pencarian Tempat Fasilitas Pelayanan Kesehatan Terdekat dan diharapkan aplikasi ini bisa bermanfaat.

2. METODE

2.1 Algoritma Dijkstra (Pencarian Jalur Terpendek)

Algoritma ini bertujuan untuk menemukan jalur terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya. Misalkan titik menggambarkan gedung dan garis menggambarkan jalan, maka algoritma Dijkstra melakukan kalkulasi terhadap semua kemungkinan bobot terkecil dari setiap titik.



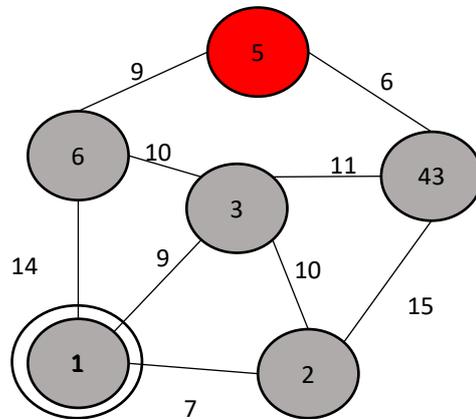
Gambar 2.1 Contoh keterhubungan antar titik dalam algoritma Dijkstra

Pertama-tama tentukan titik mana yang akan menjadi node awal, lalu beri bobot jarak pada node pertama ke node terdekat satu per satu, Dijkstra akan melakukan pengembangan pencarian dari satu titik ke titik lain dan ke titik selanjutnya tahap demi tahap. Inilah urutan logika dari algoritma Dijkstra:

1. Beri nilai bobot (jarak) untuk setiap titik ke titik lainnya, lalu set nilai 0 pada node awal dan nilai tak hingga terhadap node lain (belum terisi)
2. Set semua node “Belum terjamah” dan set node awal sebagai “Node keberangkatan”
3. Dari node keberangkatan, pertimbangkan node tetangga yang belum terjamah dan hitung jaraknya dari titik keberangkatan. Sebagai contoh, jika titik keberangkatan A ke B memiliki bobot jarak 6 dan dari B ke node C berjarak 2, maka jarak ke C melewati B menjadi $6+2=8$. Jika jarak ini lebih kecil dari jarak sebelumnya (yang telah terekam sebelumnya) hapus data lama, simpan ulang data jarak dengan jarak yang baru.
4. Saat kita selesai mempertimbangkan setiap jarak terhadap node tetangga, tandai node yang telah terjamah sebagai “Node terjamah”. Node terjamah tidak akan pernah di cek kembali, jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya.
5. Set “Node belum terjamah” dengan jarak terkecil (dari node keberangkatan) sebagai “Node Keberangkatan” selanjutnya dan lanjutkan dengan kembali ke step 3.

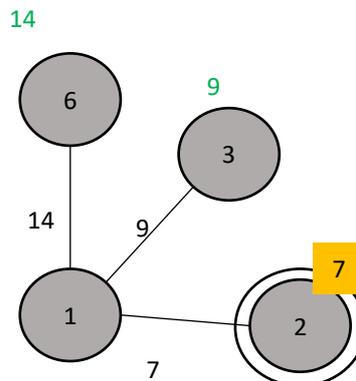
Dibawah ini penjelasan langkah per langkah pencarian jalur terpendek secara rinci dimulai dari node awal sampai node tujuan dengan nilai jarak terkecil.

1. Node awal 1, Node tujuan 5. Setiap edge yang terhubung antar node telah diberi nilai.



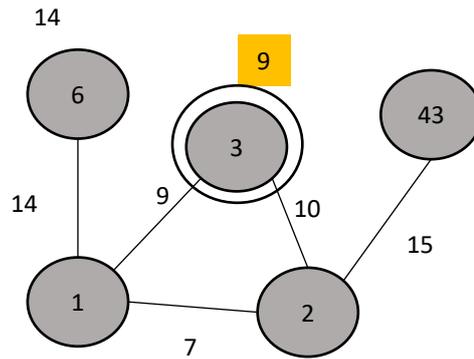
Gambar 2.2 Contoh kasus Dijkstra - Langkah 1

2. Dijkstra melakukan kalkulasi terhadap node tetangga yang terhubung langsung dengan node keberangkatan (node 1), dan hasil yang didapat adalah node 2 karena bobot nilai node 2 paling kecil dibandingkan nilai pada node lain, nilai = 7 (0+7).



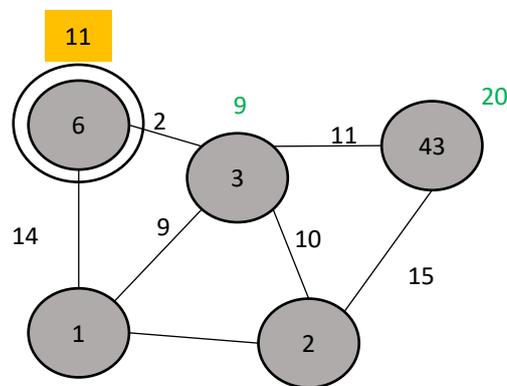
Gambar 2.3 Contoh kasus Dijkstra - Langkah 2

3. Node 2 diset menjadi node keberangkatan dan ditandai sebagai node yang telah terjamah. Dijkstra melakukan kalkulasi kembali terhadap node-node tetangga yang terhubung langsung dengan node yang telah terjamah. Dan kalkulasi Dijkstra menunjukkan bahwa node 3 yang menjadi node keberangkatan selanjutnya karena bobotnya yang paling kecil dari hasil kalkulasi terakhir, nilai 9 (0+9).



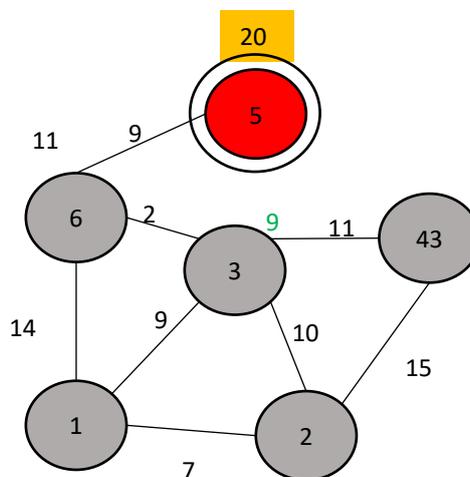
Gambar 2.4 Contoh kasus Djikstra - Langkah 3

- Perhitungan berlanjut dengan node 3 ditandai menjadi node yang telah terjamah. Dari semua node tetangga belum terjamah yang terhubung langsung dengan node terjamah, node selanjutnya yang ditandai menjadi node terjamah adalah node 6 karena nilai bobot yang terkecil, nilai $11(9+2)$.



Gambar 2.5 Contoh kasus Djikstra - Langkah 4

- Node 6 menjadi node terjamah, dijkstra melakukan kalkulasi kembali, dan menemukan bahwa node 5 (node tujuan) telah tercapai lewat node 6. Jalur terpendeknya adalah 1-3- 6-5, dan nilai bobot yang didapat adalah 20 ($11+9$). Bila node tujuan telah tercapai maka kalkulasi dijkstra dinyatakan selesai.



Gambar 2.6 Contoh kasus Djikstra - Langkah 5

2.2 Google API (Application Programming Interface)

API adalah kode program yang dapat kita integrasikan pada aplikasi atau sebuah website untuk memanfaatkan fungsi atau fitur yang disediakan Google, diantaranya adalah: (a) Language API, (b) Javascript, (c) Maps API, (d) Search API, (e) Visualization API.

2.3 GPS (Global Positioning System)

Global Positioning system (GPS) bekerja ketika sejumlah satelit yang berada di orbit Bumi memancarkan sinyalnya ke Bumi kemudian sinyal tersebut ditangkap oleh sebuah alat penerima yang nantinya diubah menjadi informasi berupa titik lokasi dari alat penerima tersebut. Alat ini bergantung penuh pada satelit maka sinyal satelit merupakan hal yang penting untuk mendapatkan informasi posisi objek yang berupa titik koordinat. Keakuratan atau ketepatan menjadi perhatian utama dalam sistem ini untuk mendapatkan sebuah lokasi atau koordinat.

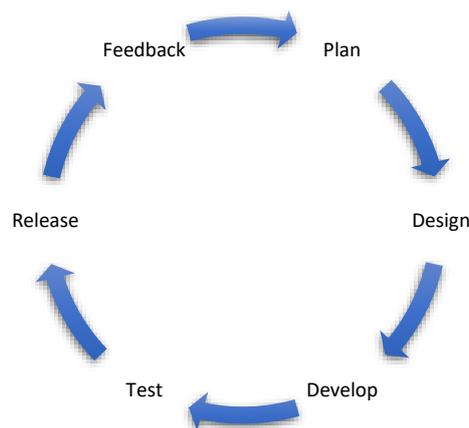
2.4 Framework

framework developer bisa mengurangi jumlah bug pada aplikasi yang dibuat. Karena, fungsi dan variabel yang sudah tersedia di dalam komponen framework. Selain itu, ada tiga fungsi framework lainnya:

- Program Lebih Terstruktur
- Praktis
- Keamanan lebih unggul

2.5 Agile Model

Sebuah “metodologi praktis” untuk memodelkan system perangkat lunak ber-teknik orientasi objek secara efektif. Metodologi AM terbentuk dari kumpulan-kumpulan praktis yang dipelihara oleh prinsip-prinsip dan nilai-nilai yang dapat diaplikasikan oleh perangkat lunak profesional dari waktu ke waktu.



Gambar 2.7 Siklus *agile* model

Tahapan khusus dalam Agile Model:

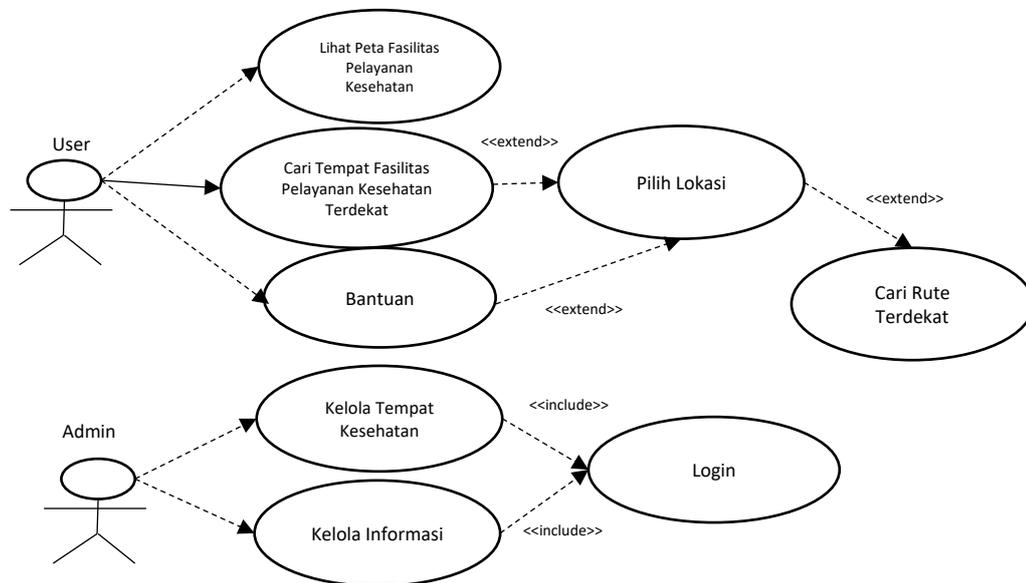
- Tahap Pemodelan (*Modeling*)
- Tahap Pengembangan (*Development*)
- Tahap Praktis (*Practice*)

3. HASIL DAN PEMBAHASAN

Algoritma Dijkstra yang diterapkan pada Aplikasi pencarian tempat fasilitas pelayanan kesehatan terdekat di Kota Depok, memudahkan masyarakat dalam mencari informasi tempat fasilitas kesehatan terdekat. Jarak dan rute tercepat yang akan dilalui juga dapat ditampilkan dalam aplikasi ini, sehingga masyarakat dapat dengan mudah mengetahui jalur tujuan tersebut.

3.1 Use Case Diagram

Use case diagram pada Gambar 3.1 dapat digunakan untuk menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem dimana masyarakat dapat melakukan pencarian tempat fasilitas pelayanan kesehatan terdekat dan admin dapat menambahkan data tempat fasilitas pelayanan kesehatan.



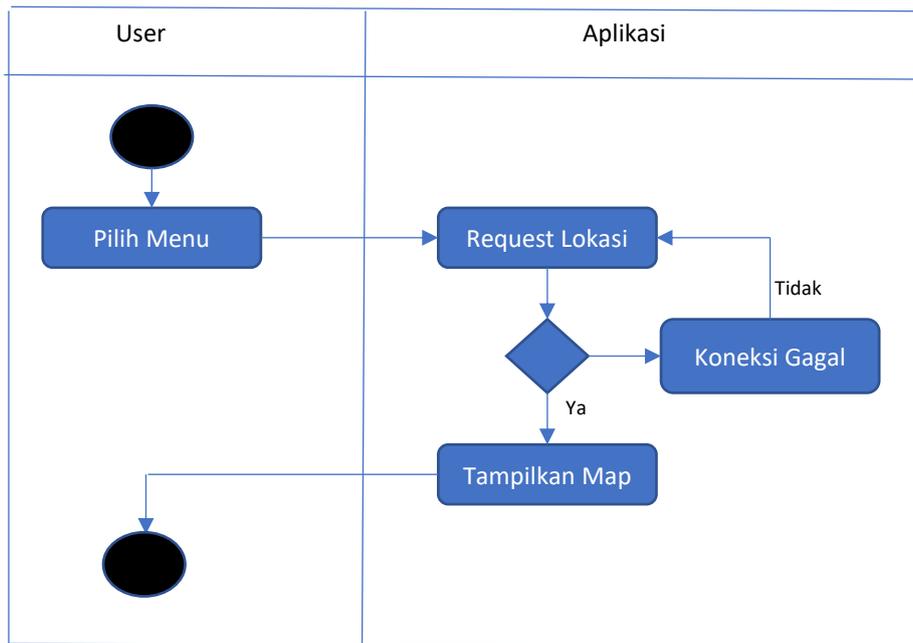
Gambar 3.1 Use Case Diagram

3.2 Diagram Activity

Diagram aktivitas atau *activity diagram* menggambarkan aliran fungsionalitas sistem. Ada beberapa diagram aktivitas dalam perancangan ini, yaitu sebagai berikut:

3.2.1. Activity Diagram User

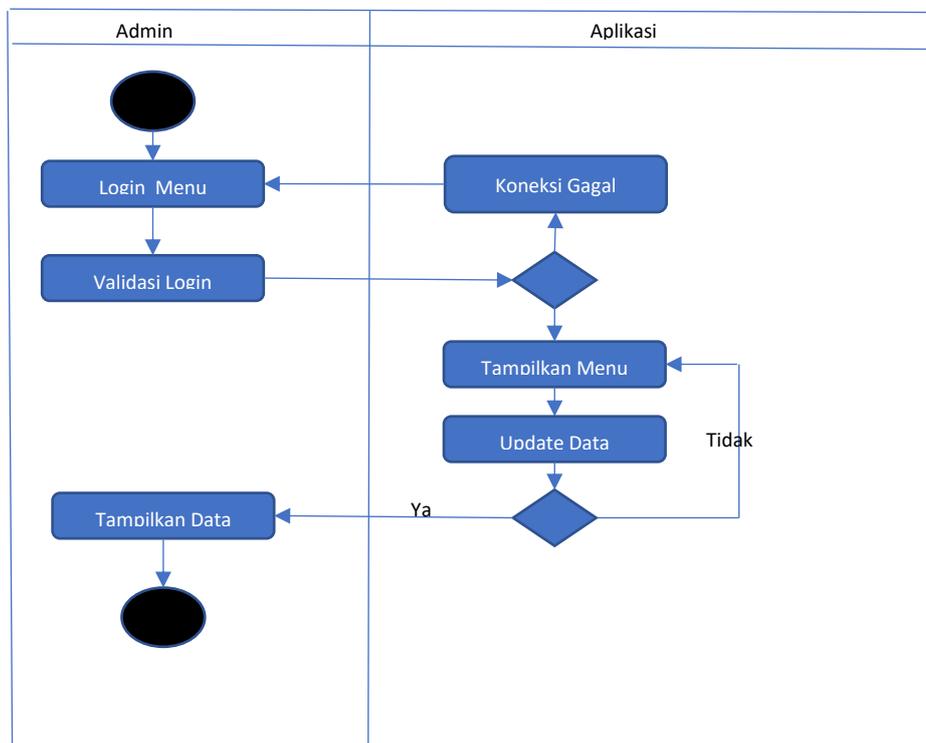
Merupakan menu pengguna dari aplikasi ini, seperti Gambar 3.2 didalamnya terdapat menu utama diantaranya peta tempat fasilitas pelayanan kesehatan, cari tempat fasilitas pelayanan kesehatan, Bantuan. User dapat memilih dengan menyentuh menu pada layar. Cari tempat fasilitas pelayanan kesehatan merupakan inti dari program, yang menampilkan data tempat fasilitas pelayanan kesehatan yang dituju, kemudian menampilkan tempat fasilitas pelayanan kesehatan yang di pilih.



Gambar 3.2.1 Activity Diagram User

3.2.2. Activity Diagram Admin

Merupakan menu admin dari aplikasi ini, didalamnya terdapat menu utama diantaranya daftar tempat fasilitas pelayanan kesehatan, dan informasi tempat fasilitas pelayanan kesehatan. Admin dapat mengupdate tempat fasilitas pelayanan kesehatan dan informasi, kemudian aplikasi akan menampilkan tempat fasilitas pelayanan kesehatan dan informasi yang sudah di *update*.



Gambar 3.2 Activity Diagram Admin

4. KESIMPULAN

Algoritma Dijkstra yang diterapkan dari hasil penelitian pada Aplikasi Pencarian Tempat Fasilitas Pelayanan Kesehatan Terdekat di Kota Depok dapat memberikan manfaat, sebagai berikut:

- Algoritma Dijkstra dapat memberikan rekomendasi rute perjalanan serta pemilihan tempat fasilitas pelayanan terdekat.
- Fitur sistem dapat memberikan rekomendasi pemilihan tempat fasilitas pelayanan terdekat dan urutan rute perjalanan dengan mempertimbangan jarak perjalanan.

5. DAFTAR PUSTAKA

- Alamsyah, and Indriani Tiara Putri. 2014. "Penerapan Algoritma Greedy Pada Mesin Penjual Otomatis (Vending Machine)." *Scientific Journal of Informatics* 1(2): 201–9.
- Ayu Fadhilah, Riki Ruli, Rakhmat. 2019. "Penanganan Gangguan Listrik Rumah Tangga Menggunakan Algoritma Greedy Untuk Penentuan Jarak Optimal." *Teknologi* 2(1): 1–21.
- Giovani, Angelina Puput et al. 2020. "Implementasi Finite State Automata Dalam Siklus Pembelajaran Magister Ilmu Komputer STMIK Nusa Mandiri." *CESS (Journal of Computer Engineering, System and Science)* 5(2): 221.
- Hartanto, Anggit Dwi et al. 2019. "Implementasi Algoritma Dijkstra Pada Game Pacman." *CCIT Journal* 12(2): 170–76.
- He, Zhenhuan. 2011. "Research on Improved Greedy Algorithm for Train Rescheduling." *Proceedings - 2011 7th International Conference on Computational Intelligence and Security, CIS 2011* (2009): 1197–1200.
- Herli, Audrey Maximillian, and Indra Kharisma Raharjana. 2015. "Sistem Pencarian Hotel Berdasarkan." 1(1): 9–16.
- Jakaria, Ribangun Bambang, and Mochamad Alfian Rosid. 2017. "Implementasi Algoritma Greedy Pada Metode Transportasi Dengan Menggunakan Vam Dalam Pendistribusian Produk." *Spektrum Industri* 15(1): 51.
- Juniar, Ahmad. 2015. "Penerapan Algoritma Greedy Pada Penjadwalan Produksi Single-Stage Dengan Parallel Machine Di Industri Konveksi." *Jurnal Sifo Mikroskil* 16(2): 175–84.
- Oktaviana, Shinta, and Abdurrahman Naufal. 2017. "Algoritma Greedy Untuk Optimalisasi Ruang dalam Penyusunan Jadwal Perkuliahan." *Multinetics* 3(1): 54.
- Patil, Bhagya M., and Basavaraj Amarapur. 2018. "Segmentation of Leaf Images Using Greedy Algorithm." *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing, ICECDS 2017* (2): 2137–41.
- Purnomo, Adi Cahyo et al. "Implementasi Algoritma Greedy Pada Layanan Taksi Wisata Berbasis Web." *Institut Teknologi Sepuluh Nopember, Kampus ITS*.
- Purwanti, Febri Anjar Pratama Desi. 2013. "Penggunaan Algoritma Dijkstra Pada Aplikasi Searching Hotel Di Kota Semarang." : 1–12.

- Stanulovic, Jelena, Nathalie Mitton, and Ivan Mezei. 2019. "Robot Task Allocation Based on Greedy-Face-Greedy Algorithm." *27th Telecommunications Forum, TELFOR 2019*.
- Sunardi, Sunardi, Anton Yudhana, and Ahmad Azhar Kadim. 2019. "Implementasi Algoritma Dijkstra Untuk Analisis Rute Transportasi Umum Transjogja Berbasis Android." *Jurnal Sistem Informasi Bisnis* 9(1): 32.
- Trisna Aryandi, Hendra Kurniawan, Muhamad Radzi Rathomi trisna.aryandi@gmail.com. "Penerapan Algoritma Greedy Dalam Menentukan Rute Terpendek Puskesmas (Studi Kasus: Kota Tanjungpinang) Trisna." : 1–10.
- Trongratsameethong, Areerat, and Jarernsri L. Mitranont. 2009. "Exhaustive Greedy Algorithm for Optimizing Intermediate Result Sizes of Join Queries." *Proceedings of the 2009 8th IEEE/ACIS International Conference on Computer and Information Science, ICIS 2009*: 463–68.
- Vimali, J. S., Sanjay Gupta, and Piyush Srivastava. 2018. "A Novel Approach for Mining Temporal Pattern Database Using Greedy Algorithm." *Proceedings of 2017 International Conference on Innovations in Information, Embedded and Communication Systems, ICIIECS 2017* 2018-Janua: 1–4.
- Wahyuningsih, Pujianti. 2018. "Penerapan Algoritma Greedy Untuk Mendeteksi Aktivitas Lansia Pada Karpas Menggunakan Arduino Mega." *Jurnal INSTEK (Informatika Sains dan Teknologi)* 3(1): 51–60.