

REKAYASA PERANGKAT LUNAK PENCARIAN RUMAH KOST DENGAN DIJKSTRA'S ALGORITHM OPTIMIZATION

Edy Nasri¹, M. Bani Husni²

Fakultas Ilmu Komputer Universitas Banten Jaya
Jl. Syeh Nawawi Albantani, Curug, Serang - Banten
Email: edynasri@unbaja.ac.id¹, banihusni@unbaja.ac.id²

ABSTRACT

Information needs are very important to help facilitate human life, one of which is the need to obtain information on temporary residence or boarding houses for students, employees and employees whose home is far away. Therefore the author designed a software engineering to make it easier for students, employees and employees to get the location and complete information about boarding houses that will be chosen to be occupied or as a reference for choosing a comfortable boarding house. Applications that are designed can show boarding locations, prices, boarding facilities, boarding rules, boarding house contacts and the fastest route, closest to the boarding house application boarding house where the authors design using Dijkstra's Algorithm Optimization and Google API to help get the shortest route to the place the boarding house intended. The website application that is designed is expected to help facilitate students, employees or entrants to get fast and complete information in searching for boarding house places.

Keywords: Boarding Application, API, Dijkstra Algorithm, Website.

PENDAHULUAN

Dengan adanya pertumbuhan penduduk yang pesat maka akan berkorelasi dengan kebutuhan tempat tinggal disertai munculnya revolusi industri 4.0 untuk kebutuhan akan pengembangan teknologi informasi yang semakin pesat segala bentuk kegiatan dan pola tingkah manusia sudah dipengaruhi oleh teknologi informasi. Manusia semakin gencar mencari cara yang dianggap paling efektif dan efisien untuk memenuhi kebutuhannya dalam mendapatkan informasi. Saat ini kebutuhan tempat tinggal sementara atau rumah kost sangat banyak dicari oleh Mahasiswa, Pekerja dan masyarakat lainnya yang belum ada tempat tinggal baik berasal dari luar kota serang maupun dalam kota serang serang itu sendiri, maka akan kebutuhan informasi yang cepat dan akurat seputar tempat rumah seperti hal lokasi, fasilitas, harga dan akses jalan secara lengkap dan mudah untuk di akses website maupun smartphone.

Dengan kebutuhan dan masalah diatas maka dibutuhkan rekayasa perangkat lunak dengan pendekatan *Dijkstra's Algorithm Optimization* untuk

memudahkan dalam melakukan pencarian dan pemesanan rumah kost agar calon penghuni rumah kost tidak perlu mencari secara langsung dan dapat mencari dengan mudah menggunakan aplikasi. *Dijkstra's Algorithm Optimization* adalah sebuah algoritma yang dipakai dalam memecahkan permasalahan jarak terpendek (*shortest path problem*) untuk sebuah graf berarah (*directed graph*). Algoritma ini merupakan salah satu algoritma yang efektif dalam memberikan lintasan terpendek dari suatu lokasi ke lokasi yang lain pada prinsip dari algoritma ini adalah dengan pencarian dua lintasan yang paling kecil.

METODE PENELITIAN

Algoritma Optimalisasi Jiksa adalah sebuah algoritma yang dipakai dalam *shortest path problem* atau memecahkan permasalahan jarak terpendek untuk *directed graph* atau sebuah graf berarah.

Algoritma ini dipublikasikan pada tahun 1959 jurnal *Numerische Mathematik* yang berjudul “*A Note on Two Problems in Connexion with Graphs*” dan dianggap sebagai algoritma greedy. Shortest path problem dari sebuah titik ke titik akhir adalah sebuah masalah yang sering timbul dalam optimasi yang banyak digunakan sebagai uji coba sebuah algoritma yang diajukan. Shortest path problem merupakan hal yang terbaik untuk mewakili masalah optimisasi, dikarenakan permasalahan yang ada mudah dimengerti (menjumlahkan seluruh edge yang dilalui) tetapi memiliki banyak opsi solusi.

Menurut Andrew Goldberg peneliti Microsoft Research Silicon Valley, mengatakan ada banyak alasan mengapa peneliti terus mempelajari masalah pencarian jalan terpendek. “Jalan terpendek adalah masalah optimasi yang relevan untuk berbagai macam aplikasi, seperti jaringan routing, game, desain sirkuit, dan pemetaan”. Deskripsi matematis untuk grafik dapat diwakilkan sebagai rumus $G = \{V,E\}$, merupakan sebuah grafik (G) kemudian didefinisikan oleh $V = \text{Vertex}$, satu set simpul dan koleksi Edge (E).

Dijkstra's Algorithm Optimization bekerja dengan membuat jalur ke satu simpul optimal pada setiap langkah. Jadi pada langkah ke n, setidaknya ada n

node yang sudah kita tahu jalur terpendek. Langkah-langkah algoritma Dijkstra dapat dilakukan dengan langkah-langkah berikut:

Pertama menentukan titik awal dari node, kemudian beri bobot jarak pada node pertama kemudian ke node terdekat dengan cara satu per satu, algoritma ini akan melakukan pembaharuan dalam pencarian dari satu titik ke titik lain sehingga ke titik selanjutnya dengan cara tahap demi tahap.

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. Semua node harus di set dimana yang belum dilalui dan set node pertama sebagai Node keberangkatan, dari node keberangkatan pertimbangkan node yang lainnya yang belum dilaluinya kemudian hitung jaraknya dari titik keberangkatan tersebut. apabila jarak lebih kecil dari jarak sebelumnya tetapi telah terekam sebelumnya hapus data yang lama, kemudian simpan ulang dari data jarak dengan jarak yang baru kemudian kita selesai mempertimbangkan setiap jarak terhadap node yang lainnya, berikan tandai node yang telah dilalui sebagai Node yang telah dilewati. Setiap node yang dilewati tidak akan pernah di cek kembali, jarak yang akan disimpan adalah jarak terakhir dan yang paling minimal atau sedikit bobotnya.

Set “Node belum dilewati” dengan jarak terkecil (dari node keberangkatan) sebagai node keberangkatan” selanjutnya dan ulangi langkah 5.

Jika dari sebuah graf melambangkan tempat-tempat serta bobot sisi dimana melambangkan jarak antara dua tempat tersebut, maka Algoritma Optimalisasi Dijkstra dapat digunakan untuk menemukan sebuah jarak yang terpendek antara dua tempat.

Input algoritma ini adalah sebuah graf berarah yang berbobot (weighted directed graph) G dan sebuah sumber vertex dalam G dan V adalah himpunan semua vertices dalam graph G . Setiap sisi dari sebuah graf merupakan pasangan vertices (u,v) dimana melambangkan sebuah korelasi dari vertex u ke v . semua himpunan E disebut tepi. Bobot atau weights dari semua sisi dihitung dengan fungsi :

$$w: E \rightarrow [0, \infty)$$

maka $w(u,v)$ adalah jarak positif dari vertex u ke vertex v .

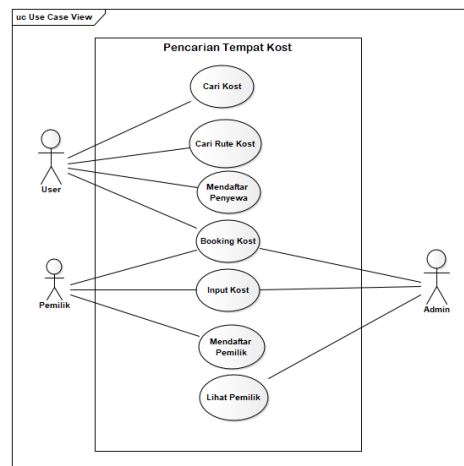
Merupakan biaya dari sebuah sisi yang dapat dianggap sebagai jarak antara dua buah vertex, merupakan jumlah jarak dari semua sisi dalam jalur tersebut, untuk sepasang vertex s dan t dalam vertex v , algoritma ini untuk menghitung jarak terpendek dari s ke t .

Dijkstra merupakan salah satu varian bentuk algoritma populer dalam pemecahan persoalan terkait masalah optimasi pencarian lintasan terpendek sebuah lintasan yang mempunyai panjang minimum dari verteks a ke z dalam grap yang berbobot, dalam bobot ini bilangan positif jadi tidak dapat dilalui oleh node yang bilangan negatif. Tetapi jika terjadi maka untuk penyelesaian yang diberikan adalah tidak terhingga atau infinity.

Pada Algoritma Optimalisasi Dijkstra, node digunakan karena Algoritma Optimalisasi Dijkstra menggunakan graph berarah untuk penentuan rute listasan terpendek.

HASIL DAN PEMBAHASAN

1. Pemodelan Sistem
- 1.2. Usecase Diagram

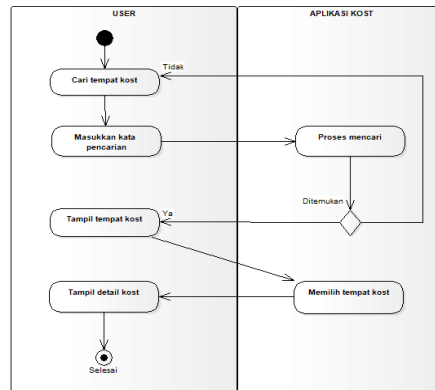


Gambar 1. Usecase Diagram Cari Rumah kost

1.3. Activity Diagram

1.3.1. Activity Diagram User Mencari Tempat Rumah Kost

User melakukan pencarian tempat kost dengan memasukkan kata pencarian lalu aplikasi kost memproses pencarian kost jika ditemukan maka akan tampil tempat kost lalu user memilih tempat kost lalu aplikasi akan menampilkan detail kost, jika tidak ditemukan ulangi ke langkah cari tempat kost.

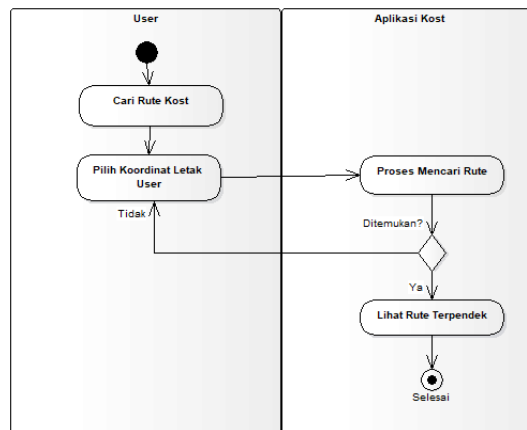


Gambar 2. Activity Diagram mencari Rumah Kost

Aktor	Keterangan
User	User membuka menu cari tempat kost, lalu memasukkan kata kunci pencarian dan aplikasi memproses pencarian tempat kost, jika di temukan maka akan tampil daftar kost sesuai keyword yang di cari lalu user memilih tempat kost dan akan tampil detail dari tempat kost yang dipilih, jika tidak maka ulangi ke proses memasukkan kata pencarian.

1.3.2. Activity Diagram User Mencari Rute Kost

Setelah menemukan tempat kost maka user akan mencari rute kost yaitu dengan pilih rute kost lalu tentukan koordinat letak user maka aplikasi akan mencari rute terdekat menggunakan algoritma djikstra jika ditemukan maka akan memperlihatkan rute nya jika tidak ditemukan ulangi dari proses pilih koordinat letak user.

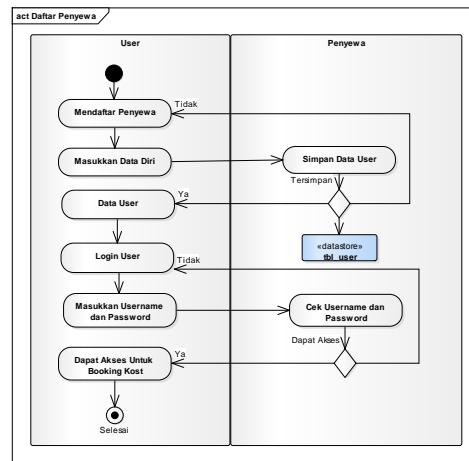


Gambar 3. Activity Diagram Cari Rute Rumah Kost

Aktor	Keterangan
User	Setelah user menemukan tempat rumahkost pilihannya maka user mencari tempat kost yaitu dengan cara memilih titik koordinat letak user lalu aplikasi akan mencari rute terdekat untuk menuju tempat kost, jika ditemukan maka rute terpendek akan di dapatkan jika tidak maka pilih/tentukan kembali koordinat letak user

1.3.3. Activity Diagram Daftar Penyewa

User melakukan daftar sebagai penyewa kost yaitu pilih menu daftar penyewa lalu isi form dengan memasukkan data diri maka aplikasi akan memproses penyimpanan data user, setelah tersimpan user dapat login ke aplikasi dengan memasukkan username dan password lalu jika username dan password benar maka akan mendapatkan akses untuk booking tempat kost.

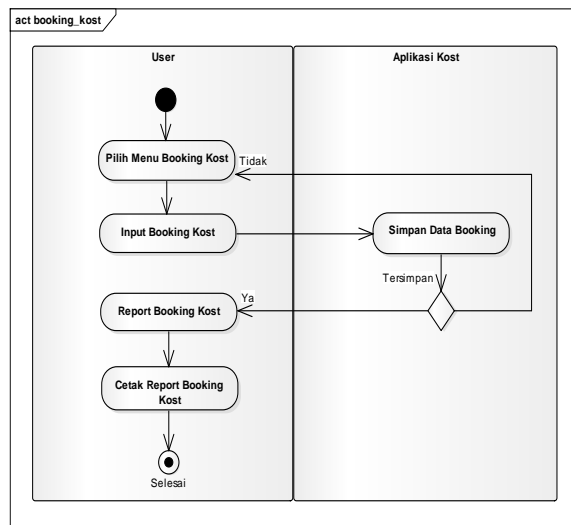


Gambar 4. Activity Diagram Mendaftar Penyewa

Aktor	Keterangan
User	Jika sudah menemukan Rute dan akan melakukan booking kost maka user harus daftar sebagai user penyewa, caranya yaitu dengan pilih menu daftar lalu masukkan data diri secara lengkap dan aplikasi akan menyimpan data user ke tabel tbl_user, lalu user melakukan login ke aplikasi dengan memasukkan username dan password yang sudah didaftarkan dan aplikasi akan cek username dan passwordnya apakah benar atau salah, jika benar maka user dapat melakukan booking kost dan jika tidak maka ulangi ke login.

1.3.4. Activity Diagram Booking Kost

Setelah user penyewa login selanjutnya pilih menu booking kost Lalu input form booking kost (pilih kost dan tanggal booking) lalu aplikasi akan menyimpan booking dan user menerima report booking untuk di cetak dan diserahkan kepada pemilik kost ketika deal sebagai bukti booking.

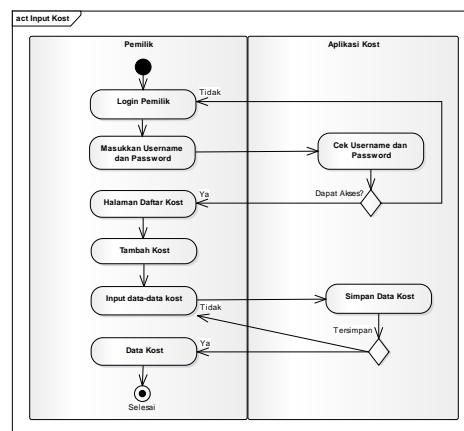


Gambar 5. Activity Diagram Booking Kost

Aktor	Keterangan
User	Setelah user berhasil login maka user dapat akses booking kost, user input booking kost lalu aplikasi akan menyimpan data booking kost lalu akan muncul report booking kost dan user cetak report booking kost

1.3.5. Activity Diagram Input Rumah Kost

Untuk pemilik kost pertama login sebagai user pemilik terdaftar dengan memasukkan username dan password lalu aplikasi akan cek apakah username dan password benar atau salah, jika benar masuk ke halaman daftar (list) data kost pilih menu tambah kost dan input form kost dengan lengkap lalu aplikasi akan menyimpan data kost.

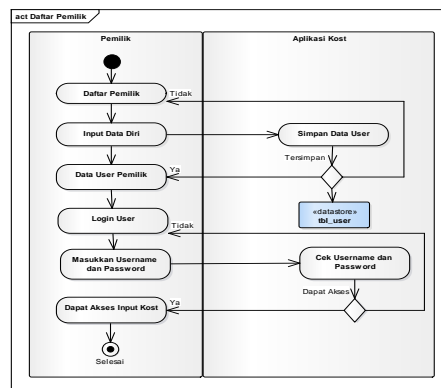


Gambar 6. Activity Diagram Input Rumah Kost

Aktor	Keterangan
User Pemilik	Pilih Menu Login lalu masukkan username dan password maka aplikasi akan mengecek apakah username dan password benar/terdaftar, jika ya dapat akses masuk kehalaman pemilik melihat menu daftar kost lalu user pemilik tambah kost dengan input data-data kost maka aplikasi akan menyimpan data kost, maka dihalaman user pemilik akan muncul data kost.

1.3.6. Activity Diagram Mendaftar Pemilik

Untuk daftar sebagai pemilik pilih menu daftar pemilik lalu input data diri secara lengkap maka aplikasi akan menyimpan data user pemilik, jika data tersimpan maka akan muncul data user yang telah diinput selanjutnya pemilik login dengan username dan password yang sudah didaftarkan sebelumnya, jika login berhasil maka mendapatkan akses untuk input kost.

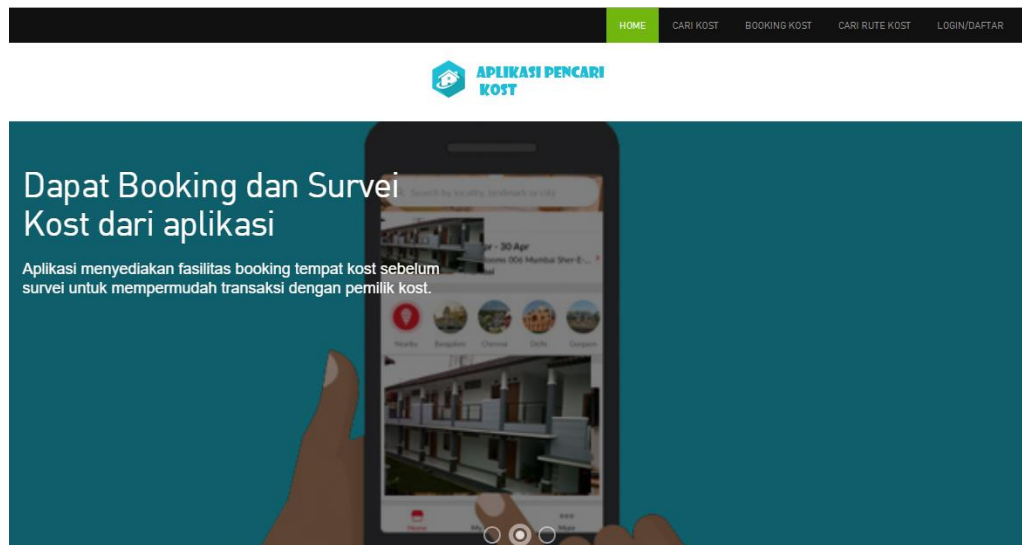


Gambar 7. Activity Diagram Mendaftar Pemilik

1.4. IMPELEMENTASI

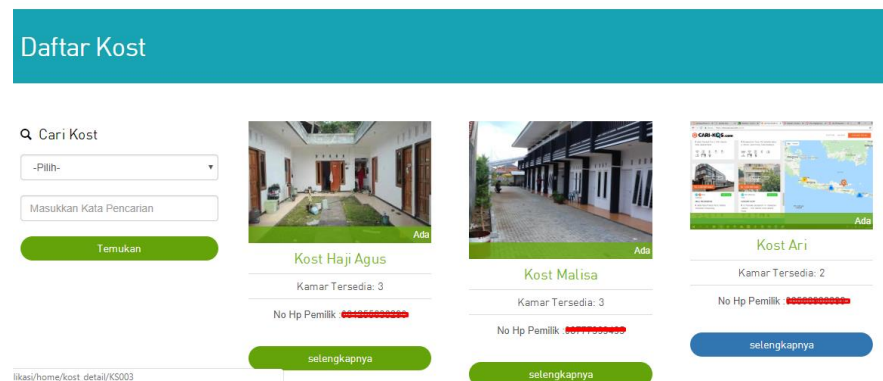
1.4.1. Tampilan menu utama,

Rekayasa Perangkat Lunak pencarian rumah kost dengan Dijkstra's Algorithm Optimization, bias digunakan di Website dan Smart Phone.



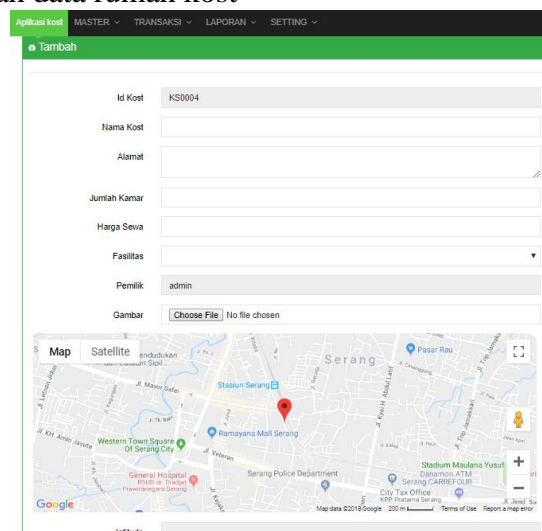
Gambar 8. Menu Utama

1.4.2. Tampilan Daftar Rumah Kost



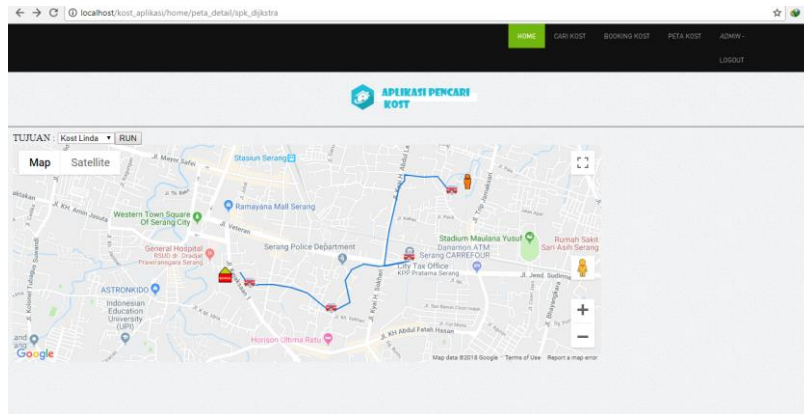
Gambar 9. Daftar Rumah Kost

1.4.3. Tampilan tambah data rumah kost



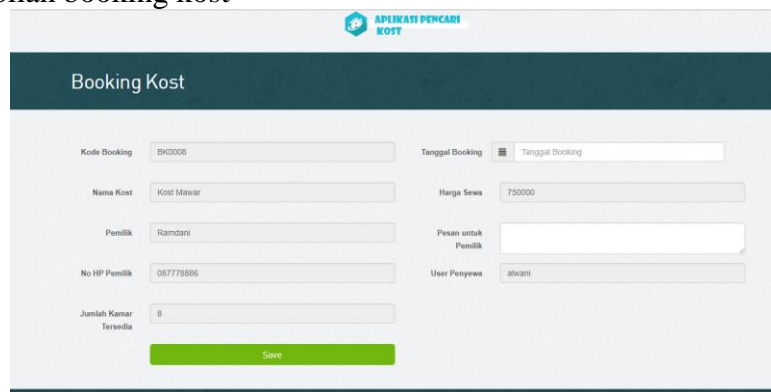
Gambar 10. Tampil untuk pendaftaran rumah kost

1.4.4. Tampilan cari rute dengan algoritma dijkstra



Gambar 11. Tampilan Cari Rute Rumah Kost

1.4.5. Tampilan booking kost



Gambar 12. Tampilan Booking Kost

KESIMPULAN

Untuk membangun Aplikasi Rekayasa Perangkat Lunak pencari tempat rumah kost dengan menggunakan metode perancangan sistem prototype dan *dijkstra's algorithm optimization* untuk algoritma penyelesaian masalahnya, dan pemrograman ini dibangun under web dengan menggunakan framework codeigniter untuk membuat aplikasi serta memanfaatkan *Google Maps Application Programming Interface*

Untuk mendapatkan rute terpendek menuju tempat kost menggunakan *dijkstra's algorithm optimization* dengan membuat node-node, jalur, tempat kost sebagai tujuannya lalu semua itu satukan dalam sebuah graph didalam aplikasi.

Membuat Pemesanan atau booking tempat rumah kost melalui aplikasi yaitu dengan mendaftar sebagai penyewa lalu pilih tempat kost yang diinginkan dan booking tempat kost tersebut dengan mengisi form booking yang disediakan.

DAFTAR PUSTAKA

- Adi Nugroho. 2009. Rekayasa Perangkat Lunak Menggunakan UML Dan Java. ANDI, Yogyakarta.
- Basuki, Awan. 2010. Membangun Web Berbasis PHP dengan Framework Codeigniter. Lokomedia: Yogyakarta.
- J.A, Blanco, D, Upton, 2009, Codeigniter 1.7, Packt Publishing: Birmingham.
- Reddy, Martin. (2011). API Design for C++. United States of America: Morgan Kaufmann
- Safaat H, Nazruddin, 2012, Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android (edisi revisi), Bandung: Informatika.
- Shelly, Cashman, Vermatt, 2009. Discovering Computers. Jakarta: Salemba Infotek.
- Subhan, Mohamad. 2012, Analisa Perancangan sistem Informasi, Jakarta: Lentera Ilmu Cendikia.
- Sutabri, Tata. 2012, Konsep Sistem Informasi. Yogyakarta: Andi
- Tulach, J. (2008). Practical API Design: Confessions of a Java Framework Architect. United States of America: Apress
- W. Pramana, Hengky. 2012. Aplikasi Inventory Berbasis Access 2003, Jakarta: PT. Elex Media Komputindo.
- Widodo Prabowo, dkk, 2011, Pemodelan Sistem Berorientasi Obyek Dengan UML, Graha ilmu, Yogyakarta.
- Whitten, Bentley, 2009, System Analysis and Design for the Global Enterprise. New York: McGraw-Hill.
- Widodo Pudjo Prabowo, Herlawati. 2011, Menggunakan UML, Informatika: Bandung
- <https://mti.binus.ac.id/2017/11/28/algorithm-dijkstra/> (diakses tanggal 07/07/2018 - 19:10)
- journals.ums.ac.id/index.php/jiti/article/download/648/389 (diakses tanggal 08/07/2018 - 19:10)
- <http://www.jurnaleeccis.ub.ac.id/index.php/eccis/article/view/214/186> (diakses tanggal 08/07/2018 - 21:10)
- <http://jurnal.umk.ac.id/index.php/simet/article/view/226> (diakses tanggal 28/04/2018 - 21:10)