SISTEM LAYANAN PENCARIAN RUTE BELANJA MENGGUNAKAN ALGORITMA DIJKSTRA

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Diajukan untuk memenuhi sebagian persyaratan mencapai gelar
Sarjana Komputer

Disusun oleh :

MUHAMMAD ABIZAR FAHRI

NIM. 115060900111038

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

UNIVERSITAS BRAWIJAYA

PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER

PROGRAM STUDI TEKNIK INFORMATIKA

MALANG

2015

LEMBAR PERSETUJUAN

SISTEM LAYANAN MANDIRI PENCARIAN RUTE BELANJA MENGGUNAKAN ALGORITMA DIJKSTRA

SKRIPSI

Untuk memenuhi persyaratan

memperoleh gelar Sarjana Komputer

Disusun oleh:

MUHAMMAD ABIZAR fAHRI

NIM. 115060900111038

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Rekyan Regasari M. P, S.T., M.T	**Hurriyatul Fitriyah, S.T, M.Sc**
NIK. 7041406120253	**NIK. 2013048510012001**

LEMBAR PENGESAHAN

SISTEM LAYANAN MANDIRI PENCARIAN RUTE BELANJA MENGGUNAKAN ALGORITMA DIJKSTRA

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun oleh :

MUHAMMAD ABIZAR FAHRI

NIM. 115060900111038

Skripsi ini telah diuji dan dinyatakan lulus pada
tanggal 05 Agustus 2015

Penguji I

Fitri Utaminingrum, Dr. Eng, S.T, M.T., M.Eng.

NIP. 19820710 200812 2 001

Penguji II

Barlian Henryranu P, S.T., M.T.

NIK. 821024 06 2 2 0254

Penguji III

Wijaya Kurniawan, S.T., M.T.

NIK. 820125 16 1 1 0418

Mengetahui

Ketua Program Studi Teknik Informatika

Drs. Marji, M.T.

NIP. 19670801 199203 1 001

ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah SKRIPSI ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku. (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 05 Agustus 2015

Mahasiswa,

Muhammad Abizar Fahri

NIM. 115060900111038

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa, karena Rahmat-Nya lah penulis dapat menyelesaikan skripsi ini. Adapun maksud penyusunan skripsi ini adalah untuk memenuhi salah satu persyaratan dalam menempuh ujian Sarjana Fakultas Ilmu Komputer. Judul skripsi yang disusun adalah: **"Sistem Layanan Mandiri Pencarian Rute Belanja Menggunakan Algoritma Dijkstra"**

Banyak kesulitan dan hambatan yang dialami oleh penulis dalam menyusun skripsi ini terutama dalam mendapatkan data dan mengolahnya, tetapi semua itu telah dapat diatasi dengan baik berkat dukungan dan bantuan dari berbagi pihak. Untuk itulah pada kesempatan ini penulis mengucapkan terima kasih kepada:

1) Bapak dan Ibu yang saya cintai, yang telah memberikan dukungan, semangat serta doa.
2) Bapak Adharul Mutaqqin, S.T., M.T. selaku Ketua Program Studi Sistem Komputer Universitas Brawijaya Malang.
3) Ibu Rekyan Regasari Mardi Putri, S.T., M.T. selaku dosen pembimbing I yang telah memberikan pengarahan dalam pembuatan proposal skripsi ini.
4) Ibu Hurriyatul Fitriyah,S.T., M.Sc selaku dosen pembimbing II yang telah memberikan saran-saran yang sangat berguna dalam penyelesaian proposal skripsi ini.
5) Barlian Henryranu, S.T., M.T. selaku Kepala Laboratorium Sistem Komputer dan Robotika Program Teknologi Informasi dan Ilmu Komputer.
6) Aninditya Nugroho, S.T (Mas Didit) selaku laboran Kepala Laboratorium Sistem Komputer dan Robotika Program Teknologi Informasi dan Ilmu Komputer yang selalu membantu & menyediakan alat-alat skripsi untuk penulis.

7) Segenap dosen Sistem Komputer dan Fakultas Ilmu Komputer Universitas Brawijaya atas segenap ilmu pengetahuan dan perhatian yang diberikan Segenap staff dan pegawai Fakultas Ilmu Komputer Universitas Brawijaya atas segala bantuan yang bersifat administratif.
8) Andri dan Bagus yang telah banyak memberi masukan dalam pengerjaan skripsi ini.
9) Ricus yang ingin Ami, Mas Yannuar, Bagas dan seluruh isi ngopirasi dimasukkan kedalam halaman ini.
10) Seluruh teman-teman angkatan 2011 dan pihak yang telah membantu yang tidak dapat saya sebutkan satu per satu.

Penulis menyadari bahwa proposal skripsi ini jauh dari sempurna, oleh karena itu untuk segala kritik dan saran yang membangun penulis ucapkan terima kasih. Penulis mengharapkan semoga proposal skripsi ini dapat berguna bagi yang membutuhkannya.

Malang, Agustus 2015

Penulis

repository.ub.ac.id

ABSTRAK

Belanja telah menjadi satu kegiatan yang tidak terlepaskan dari masyarakat modern. Rata-rata kegiatan belanja menghabiskan waktu 1,4 jam setiap harinya dimana sebagian besar waktu habis untuk berkeliling rak mencari barang yang diinginkan. Maka diperlukan sebuah sistem yang memberikan layanan pencarian agar kegiatan belanja dapat dilakukan secara efisien. Sebuah aplikasi Intelligent Self Service System mengggunakan Delphi dan Mysql dirancang untuk mencari rute belanja terpendek. Proses pencarian rute terpendek dilakukan menggunakan algoritma Dijsktra. Pada sistem ini, pengguna memasukkan daftar belanja melalui aplikasi. Kemudian *server* menentukan rute terpendek belanja dimana rute ini dikirimkan ke modul keranjang belanja menggunakan Zigbee Pro Series 2 secara unicast dibawah protokol zigbee. Modul pada keranjang belanja terdiri dari Arduino Uno dan RFID scanner RDM6300 dimana hasil rute tersebut ditampilkan pada LCD 16x2 bit. Hasil pengujian membuktikan sistem dapat memberikan rute sesuai metode yang dirancang dengan akurasi 100% dan sistem mampu memberikan rute dengan jarak terpendek berdasarkan total *cost* dengan akurasi 67%.

Kata kunci : *RFID, Djiktra, Zigbee, Belanja, Rute.*

UNIVERSITAS BRAWIJAYA

repository.ub.ac.id

ABSTRACT

Shopping has become a must be activity from modern society. On average shopping activities spent 1.4 hours per day in which most of the time spent on shelves around looking for the desired item. It needs a system that provides search services so that the activities of shopping can be done efficiently. An application Intelligent Self-Service System using Delphi and MySQL wass designed to find the shortest route shopping. The shortest route search process was done using Dijsktra Algorithm. On the Intelligent Self-Service System, the user input the shopping list to application. Then the server determines the shortest route where the shopping list idelivered to the shopping cart module using Zigbee Pro Series 2 under ZigBee protocol unicastly. Modules in the shopping basket consists of Arduino Uno and RFID scanners RDM6300 where the results are displayed on the LCD 16x2 bit. The test results prove the system can provide appropriate service methods designed with 100% accuracy, and the system is able to provide a more efficient service with an accuracy of 67%.

Keywords : RFID, Djikstra, Zigbee, Shopping, Route.

UNIVERSITAS BRAWIJAYA

DAFTAR ISI

KATA PENGANTAR i
ABSTRAK iii
ABSTRACT iv
DAFTAR GAMBAR vii
BAB I 1
PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 2
1.3 Tujuan Penelitian 3
1.4 Manfaat Penelitian 3
1.5 Batasan Penelitian 3
1.6 Sistematika Penulisan 3
BAB II 5
TINJAUAN PUSTAKA DAN DASAR TEORI 5
2.1 Tinjauan Pustaka 5
2.2 Dasar Teori 6
2.2.1 *Intelligent Self Service System (*ISSS*)* 6
2.2.2 Teori Graf 7
2.2.3 Algoritma Dijkstra 9
2.2.4 Arduino Uno 10
2.2.5 Modul Wireless Radio Frequency 2,4 GHz 14
2.2.8 MySQL 19
BAB III 20
METODE PENELITIAN 20
3.1 Studi Literatur 20
3.2 Analisis Kebutuhan Sistem 20
3.3 Perancangan Sistem 21
3.4 Implementasi Sistem 22
3.5 Pengujian dan Analisis 23
3.6 Kesimpulan 24

BAB IV 25
PERANCANGAN DAN IMPLEMENTASI SISTEM 25
4.1 Perancangan 25
4.1.1 Analisa Kebutuhan 25
4.1.2 Diagram Blok Perancangan Sistem 27
4.1.3 Perancangan Perangkat Keras 27
4.1.4 Perancangan Perangkat Lunak 29
4.2 Implementasi 33
4.2.1 Implementasi Perangkat Lunak 33
4.2.2 Implementasi Antar Muka 40
BAB V 45
PENGUJIAN DAN ANALISIS 45
5.1 Pengujian dan Analisis Algoritma RFID Scanner 45
5.2 Pengujian dan Analisis Hasil Rute 47
5.3 Pengujian dan Analisis Akurasi Rute Hasil 51
5.4 Pengujian dan Analisis Komunikasi 56
BAB VI 60
KESIMPULAN DAN SARAN 60
6.1 Kesimpulan 60
6.2 Saran 60
DAFTAR PUSTAKA 61

DAFTAR GAMBAR

Gambar 2. 1. diagram ISSS 7
Gambar 2. 2 tipe-tipe graf 8
Gambar 2. 3 Arduino Uno 12
Gambar 2. 4 Bagian Arduino Uno board 12
Gambar 2. 5 tampilan xbee 14
Gambar 2. 6 Pin Xbee S2 15
Gambar 2. 7 Tampilan RDM6300 16
Gambar 2. 8 Dimensi RDM6300 18
Gambar 3. 1 Diagram Blok Metode Penelitian 20
Gambar 3. 2 Diagram blok sistem 22
Gambar 4. 1 Desain Perangkat Keras 28
Gambar 4. 2 Diagram Alir Proses Kerja 28
Gambar 4. 3 Desain Pemetaan Lokasi 30
Gambar 4. 4 Proses Pemetaan Jarak 31
Gambar 4. 5 Perancangan Pengolahan Data pada Server 32
Gambar 4. 6 *Source Code* program *scan RFID* 34
Gambar 4. 7 *Source Code* program proses pemilihan barang belanja 36
Gambar 4. 8 *Source Code* Penentuan Rute 39
Gambar 4. 9 Tampilan Halaman *Login* 40
Gambar 4. 10 Tampilan halaman utama 41
Gambar 4. 11 Halaman keranjang belanja 42
Gambar 4. 12 Tampilan Halaman Hasil 43
Gambar 4. 13 Tampilan Halaman awal Pengaturan Admin 43
Gambar 4. 14 Tampilan Halaman Penambahan user admin 44
Gambar 5. 1 Pengujian 1 algoritma RFID Scanner 45
Gambar 5. 2 Pengujian 3 algoritma RFID Scanner 46
Gambar 5. 3 Pengujian 4 algoritma RFID Scanner 46
Gambar 5. 4 Pengujian 5 algoritma RFID Scanner 47
Gambar 5. 5 Tampilan Pengiriman Data dari Server ke Arduino 57

DAFTAR TABEL

Tabel 2. 1 Spesifikasi Arduino Uno 11
Tabel 2. 2 Spesifikasi Xbee Series 2 15
Tabel 2. 3 spesifikasi modul RDM6300.....17
Tabel 4. 1 Rancangan Paket Data dari Arduino ke Komputer.....29
Tabel 4. 2 Rancangan Paket Data dari Komputer ke Arduin.....29

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

BAB I
PENDAHULUAN

1.1 Latar Belakang

Belanja telah menjadi salah satu kegiatan yang rutin dilakukan oleh masyarakat modern. Menurut Bonnie Swaine dkk (2014) belanja adalah aktivitas penting untuk semua orang dan dapat berkontribusi signifikan kepada kepuasan seseorang dalam kehidupan. Hal ini didukung dengan semakin menjamurnya pusat perbelanjaan di pusat kota. Selama beberapa tahun terakhir riset tentang manajemen tempat belanja telah banyak dilakukan. Peningkatan kenyamanan konsumen dalam berbelanja menjadi fokus utama dalam penelitian ini.

Belanja menjadi salah satu kegiatan sehari-hari yang banyak menghabiskan waktu. Berdasarkan hasil survei yang dilakukan Badan Tenaga Kerja Amerika Serikat, rata-rata manusia menghabiskan 1,4 jam setiap hari untuk berbelanja. Sebagian besar waktu yang dihabiskan konsumen adalah mencari lokasi barang belanja yang diinginkan (Ali, 2013).

Pada pusat perbelanjaan umumnya, barang belanja disusun berdasarkan rak-rak dimana setiap rak dikategorikan untuk barang-barang sejenis. Semakin besar pusat perbelanjaan maka akan semakin banyak pula variasi produk. Hal ini juga berbanding lurus dengan bertambahnya jumlah rak yang disusun. Mencari barang dengan cara berkeliling adalah salah satu metode yang paling sering dilakukan. Pencarian secara manual ini tentu saja membutuhkan banyak waktu karena konsumen harus berkeliling rak secara acak. Terlebih jika konsumen baru pertama kali berada di pusat perbelanjaan tersebut dan tidak mengetahui pembagian rak, maka waktu belanja akan semakin banyak..

Kepuasan konsumen menjadi penting bagi pedagang karena berhubungan dengan kegiatan rekursif dimana pengalaman sebelumnya berpengaruh terhadap keputusan belanja selanjutnya. Sehingga kemudahan dalam belanja telah menjadi poin penting dalam peningkatan kepuasan konsumen. Adanya pelayanan yang membantu kebutuhan konsumen saat proses belanja menjadi hal yang penting untuk meningkatkan kepuasan belanja konsumen.

Sebuah layanan untuk membantu proses belanja telah dilakukan oleh perusahaan Tesco menghasilkan sebuah aplikasi belanja “sat-nav”. Saat program

dijalankan aplikasi akan menampilkan peta toko beserta lokasi barang belanjaan yang telah dipilih. Namun aplikasi ini masih bersifat *prototype* dan hanya bisa dijalankan pada perangkat android versi 2.2 keatas. Selain itu pada sisi keamanan aplikasi meminta informasi pengguna dan akses *wi-fi* perangkat pengguna.

Dengan adanya permasalahan tersebut perlu adanya sebuah sistem pencarian barang belanja yang lebih fleksibel digunakan bagi konsumen. Sistem memberikan layanan yang sama dengan menampilkan lokasi barang belanjaan yang telah dipilih dengan pertimbangan rute terpendek.

Berdasarkan penjelasan di atas dapat ditarik sebuah hipotesis yaitu dibutuhkan sebuah sistem pencarian posisi barang belanja yang diaplikasikan pada pusat perbelanjaan besar menggunakan algoritma dijkstra yang dapat ditanamkan pada keranjang belanja. Hasil yang ditampilkan berupa saran rute terpendek dari keseluruhan daftar lokasi barang belanja. Rute terpendek didapatkan dengan terlebih dahulu dilakukan pemetaan seluruh rak dan posisi barang belanja berdasarkan pertimbangan *cost* berupa jaraknya menggunakan algoritma dijkstra. Hasil pemetaan berupa tabel prioritas rute rak.

Dengan *id identificator* konsumen nantinya dapat mengintergrasikan pilihan belanja yang dilakukan di beberapa lokasi seperti rumah, ponsel dan lainnya menggunakan aplikasi desktop untuk memilih barang belanjaan. Pada keranjang belanja di pusat perbelanjaan dilengkapi modul *Radio Frequency* untuk berkomunikasi dengan server dan memperoleh rute terpendek yang disarankan. Dengan adanya sistem ini maka diharapkan mampu menunjang kemudahan dan keefisiensian waktu dalam melakukan kegiatan belanja.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, dapat dirumuskan beberapa masalah sebagai berikut:

1. Bagaimana mencari rute terpendek untuk memperoleh seluruh barang belanja?
2. Bagaimana akurasi pada pemilihan rute terpendek?
3. Bagaimana membuat komunikasi *wireless* antara modul dikeranjang belanja dengan *server* ?

1.3 Tujuan Penelitian

Tujuan dari perancangan sistem pencarian barang belanja adalah membuat sistem *wireless* yang dapat membantu konsumen mencari lokasi barang belanja secara lebih efisien melalui perhitungan rute belanja terpendek.

1.4 Manfaat Penelitian

Manfaat dari penelitian diharapkan memberikan sistem layanan belanja mandiri dimana konsumen dapat mencari lokasi barang belanja dengan cepat sehingga mempersingkat waktu yang dihabiskan untuk belanja.

1.5 Batasan Penelitian

Agar pembahasan tidak melebar dari latar belakang dan terfokus dengan apa yang berkaitan dengan sistem, maka batasan dalam penelitian ini yaitu:

1. Pembahasan terbatas pada perancangan hardware pada keranjang belanja yang terintegrasi dengan server.
2. Terbatas pada bagaimana modul *wireless* dapat berkomunikasi dengan *server* dengan sistem *single hop*.
3. Terbatas pada bagaimana sistem bisa memberikan saran rute terpendek menuju barang belanjaan yang diinginkan tanpa mempertimbangkan usabilitas alat secara maksimal.
4. Manajemen data dari berbagai sistem menggunakan *database* MySQL.
5. Menggunakan Xbee Pro Series 2 sebagai media transfer data.
6. Membuat interface aplikasi desktop dengan memanfaatkan Borland Delphi 2010.
7. Nilai jarak dan posisi bersifat statis dan sudah ditentukan

1.6 Sistematika Penulisan

Penulisan skematik pembahasan dan penyusunan laporan penelitian dapat diuraikan sebagai berikut:

BAB I PENDAHULUAN

Pada Bab I Pendahuluan dijelaskan mengenai latar belakang, rumusan masalah, tujuan, manfaat penelitian, batasan penelitian dan sistematika pembahasan dari Sistem Layanan Mandiri Pencarian Rute Belanja Menggunakan Algoritma Dijkstra.

BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI

Pada bab ini berisi beberapa dasar teori dan referensi penelitian yang pernah ada yang memiliki tujuan dan perancangan yang hampir sama dengan "Sistem Layanan Mandiri Pencarian Rute Belanja Menggunakan Algoritma Dijkstra".

BAB III METODE PENELITIAN

Pada bab ini membahas beberapa langkah kerja yang akan dilakukan dalam penelitian, diantaranya studi literatur, analisis kebutuhan sistem, implementasi dan pengujian sistem dalam mengintegrasikan sistem.

BAB IV PERANCANGAN DAN IMPLEMENTASI

Bab ini membahas perancangan, analisis kebutuhan dan pengujian, serta implementasi sistem dalam mengintegrasikan hardware dengan server sehingga dapat menunjukkan bahwa sistem dapat diimplementasikan dengan baik.

BAB V PENGUJIAN DAN ANALISIS

Pada bab pengujian dan analisis ditunjukkan hasil pengujian dari perancangan sistem dan aplikasi serta dijelaskan pula teknik pengujian yang dilakukan sehingga dapat menunjukkan bahwa sistem dapat diimplementasikan dengan baik.

BAB VI PENUTUP

Bab ini berisi kesimpulan dari pembahasan rumusan masalah yang ada berdasarkan analisis dan pengujian serta berisi saran-saran dan pengembangan yang dapat dilakukan dari "Sistem Layanan Mandiri Pencarian Rute Belanja Menggunakan Algoritma Dijkstra".

BAB II
TINJAUAN PUSTAKA DAN DASAR TEORI

Bab ini akan menjelaskan tinjauan pustaka yang berisi penelitian-penelitian sebelumnya yang memiliki tujuan atau metode yang hampir sama dengan penelitian yang dilakukan penulis serta berisi dasar teori yang mendukung dalam proses penelitian "Sistem Layanan Mandiri Pencarian Rute Belanja Menggunakan Algoritma Dijkstra".

2.1 Tinjauan Pustaka

Pada penelitian yang berjudul "*RFID Based Smart Shopping and Billing* ", unit komunikasi data pada sistem menggunakan modul *Wireless Zigbee Module* diletakkan pada keranjang belanja sebagai node dilengkapi LCD, IR Decoder dan modul zigbee. Selain itu terdapat modul kendali utama (*main controller*) untuk mengatur data yang masuk. Setiap item belanja memiliki RFID tag masing-masing yang terhubung dengan jumlah stok dalam database. Saat pembeli mengambil barang belanja secara otomatis query database dijalankan untuk menampilkan update billing pada keranjang belanja. Analisis menghasilkan kesimpulan bahwa dalam level pengembangan *prototype* alat mudah digunakan namun sistem komunikasi masih memiliki. Saat ada zigbee lain dengan frekuensi yang sama, tranmisi data dengan mudah diinterupsi. (Ali, 2013)

Sebuah penelitian berjudul *"Item-level RFID for Enhancement of Customer Shopping Experience in Apparel Retail"* mengenai implementasi teknologi RFID yang ditanam dalam manajemen toko. Tujuan sistem ini adalah untuk memberikan kenyamanan pelayanan belanja. Pada sistem ini modul RFID digunakan untuk menangkap kebiasaan dan kesukaan belanja dari konsumen. Sistem dikembangkan menggunakan algoritma *fuzzy screening* untuk memberikan saran penempatan lokasi barang dan promo barang-barang terbaru berdasarkan kebiasaan konsumen dan data sejarah belanja yang terakumulasi pada database. Berdasarkan analisis disimpulkan bahwa sistem bekerja secara efektif dalam melakukan penempatan item dengan biaya yang dapat diterima dan pemeliharaan yang mudah (Choi, 2015).

Penelitian berjudul Implementasi Algoritma Dijkstra Dalam Aplikasi Untuk Menentukan LintasanTerpendek Jalan Darat Antar Kota Di Sumatera Bagian Selatan dilakukan untuk melakukan pemetaan tiap titik di provinsi Sumatera Selatan untuk mendapatkan nilai terkecil menuju titik tujuan. Algoritma dijkstra dilakukan untuk mendapatkan jarak terdekat pada satu kota ke kota lain pada daerah Sumatera Bagian Selatan. Hasil pemetaan ditempatkan pada fasilitas-fasilitas umum seperti terminal bus. Berdasarkan hasil penelitian diketahui algoritma dijkstra dapat digunakan untuk mencari rute terpendek secara maksimal (Fitria, 2013).

Berdasarkan beberapa penelitian di atas, penulis membuat sebuah sistem layanan mandiri pencarian rute belanja menggunakan algoritma dijkstra. Sistem berbasis RFID sebagai verifikasi pengguna dan *Zigbee Pro Series 2* sebagai komunikasi antara modul dan server. Aplikasi *Virtual Shopping* digunakan sebagai antarmuka pengguna dalam memilih barang belanja.

2.2 Dasar Teori

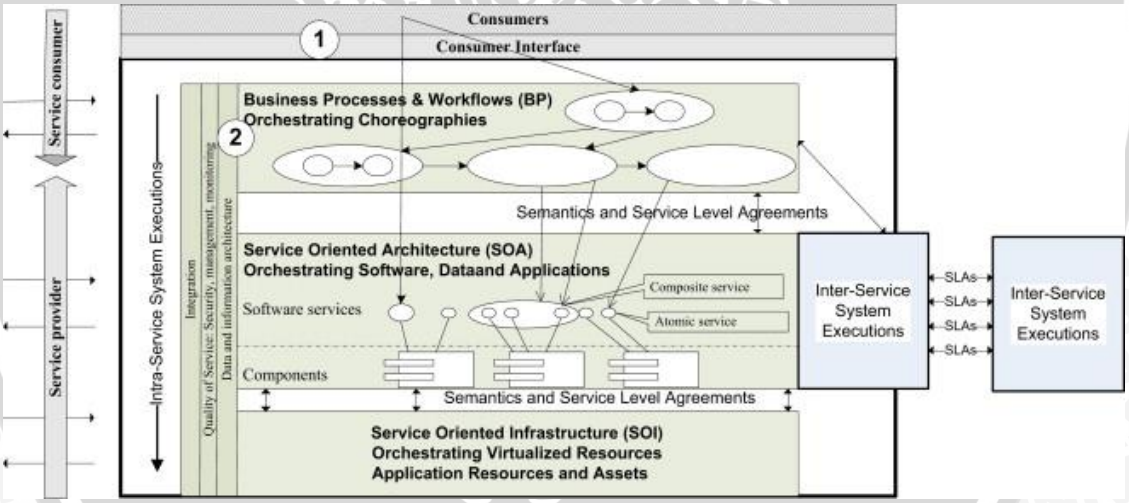
2.2.1 *Intelligent Self Service System (*ISSS*)*

Dalam konteks pelayanan, *self-service system* untuk belanja *online* adalah salah satu aplikasi yang paling cepat berkembang pada abad ke 21. Ada sebuah pertumbuhan besar dalam *self-service* yang membuat konsumen mengambil peran dari pekerja layanan dalam tujuan melayani. Sistem *self-service* terlihat menjadi trend terkini, dimana pekerjaan menjadi sedhana dengan cukup menekan tombol. *Self-checkout* merupakan salah satu teknologi *self-service* yang paling luas tersebar. Dengan sistem ini, pengguna bisa melakukan *scan*, bungkus, dan bayar barang belanjaannya sendiri. Dengan ada sekitar lebih dari 60 Milyar transaksi pada supermarket, potensi penghematan ditemukan sangat besar denagn penggunaan aplikasi *self-service (*Atkinson, 2005*)*.

Kemajuan dunia IT memunculkan sistem cerdas, aplikasi web, *cloud computing*, aplikasi *mobile*, RFID, jaringan sosial, komputasi performa tinggi, kecepatan tinggi komunikasi glbal, serta teknologi sensor dan analisis data. Teknologi ini memberikan kesempatan untuk memanajemen

teknologi kedalam hubungan pelayanan dengan mengatur hubungan sistem *self-service* dengan orang, proses, teknologi, organisasi, informasi, bahasa, hukum, aturan, matriks, pengukuran, model, dll untuk menciptakan nilai baru antara penyedia dan penerima. *Cloud* memfasilitasi ISSS, bersama dengan kamera dan sensor, kecepatan data, dan aplikasi *mobile* mendemonstrasikan potensi tak terhingga untuk menciptakan layanan cerdas otomatis tingkat tinggi (Demirkhan, 2014).

Prinsip dasar dari *Intelligent Self-Service Systems (*ISSS*)* adalah mengatur penawaran pelayanan dengan konsumen dan memecahkan struktur proses bisnis menjadi model layanan mandiri. Pada gambar 2. Ditunjukkan ISSS meliputi konsep model dari intra dan inter organisasi dari proses bisnis, layanan dan arsitektur pelaksanaan dengan *cloud-service,* rantai suplai organisasi, layanan cerdas *mobile* , kecepatan transfer data memungkinkan sistem bisnis cerdas.

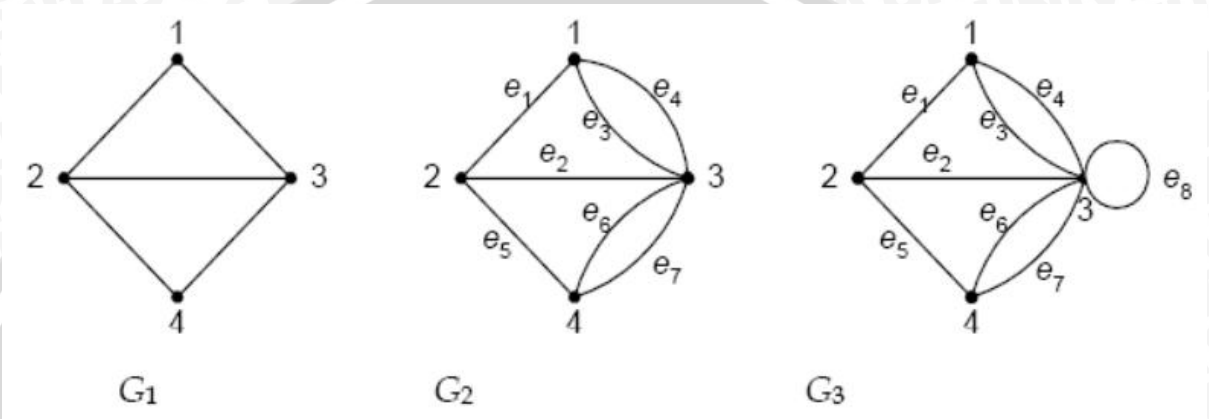


Gambar 2. 1. diagram ISSS
(Sumber : Demirkhan, 2014)

2.2.2 Teori Graf

Graf 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 [1]. Simpul pada graf dapat dinomori dengan huruf, seperti a, b, c...dst, dengan bilangan asli 1, 2, 3...dst, atau gabungan keduanya. Sedangkan sisi yang menghubungkan

simpul u dengan simpul v dinyatakan dengan pasangan (u, v) atau dinyatakan dengan lambang e1, e2....en 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 dwimatra yang dihubungkan dengan sekumpulan garis (sisi)



Gambar 2. 2 tipe-tipe graf
(Sumber: Fitria,2013)

Gambar di atas memperlihatkan tiga buah graf, G1, G2 dan G3 . G1 adalah graf dengan himpunan simpul V dan himpunan sisi E adalah: V = {1, 2, 3, 4} E = {(1, 2), (1, 3), (2, 3), (2, 4), (3, 4)} G2 adalah graf dengan himpunan simpul V dan himpunan sisi E adalah: V = {1, 2, 3, 4} E = {(1, 2), (2, 3), (1, 3), (1, 3), (2, 4), (3, 4), (3, 4)} = {e1, e2, e3, e4, e5, e6, e7} G3adalah graf dengan himpunan simpul V dan himpunan sisi E adalah: V = {1, 2, 3, 4} E = {(1, 2), (2, 3), (1, 3), (1, 3), (2, 4), (3, 4), (3, 4), (3, 3)} = {e1, e2, e3, e4, e5, e6, e7, e8} Pada G2, sisi e3 = (1, 3) dan sisi e4= (1, 3) dinamakan sisi-ganda (*multiple edges* atau *parallel edges*) karena kedua sisi ini menghubungi dua buah simpul yang sama, yaitu simpul 1 dan simpul 3. Pada G3 , sisi e8 = (3, 3) dinamakan gelang atau kalang (*loop*) karena ia berawal dan berakhir pada simpul yang sama.

2.2.2.1 Graf Berbobot (*Weighted Graph*)

Graf berbobot adalah graf yang setiap sisinya diberi sebuah harga (bobot) [1]. Bobot pada tiap sisi dapat berbeda-beda bergantung pada

masalah yang dimodelkan dengan graf. Bobot dapat menyatakan jarak antara dua buah kota, biaya perjalanan antara dua buah kota, waktu tempuh pesan (*message*) dari sebuah simpul komunikasi ke simpul komunikasi lain (dalam jaringan komputer) ,ongkos produksi dan sebagainya.

2.2.2.2 Lintasan Terpendek

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 kita gunakan di sini adalah bahwa semua bobot bernilai positif. Kata terpendek berbeda-beda maknanya bergantung pada tipikal persoalan yang akan diselesaikan. Namun, secara umum terpendek berarti meminimisasi bobot pada suatu lintasan dalam graf (Fitria, 2013).

2.2.3 Algoritma Dijkstra

Algoritma yang ditemukan oleh *Dijkstra* untuk mencari path terpendek merupakan algoritma yang lebih efisien dibandingkan algoritma *Warshall*, meskipun implementasinya juga lebih sukar. Misalkan G adalah graf berarah berlabel dengan titik-titik V(G) = {v1, v2,..., vn} dan path terpendek yang dicari adalah dari v1 ke vn. Algoritma *Dijkstra* dimulai dari titik v1. dalam iterasinya, algoritma akan mencari satu titik yang jumlah bobotnya dari titik 1 terkecil. Titik-titik yang terpiih dipisahkan dan titik-titik tersebut tidak diperhatikan lagi dalam iterasi berikutnya.

Misalkan:

```
1.  V (G) = {v1, v2,.., vn}
2.  L = Himpunan titik-titik ε V (G) yang sudah terpilih dalam jalur
    path terpendek.
3.  D(j) = Jumlah bobot path terkecil dari v1 ke vj.
4.  w(i,j) = Bobot garis dari titik vi ke vj.
    w*(1,j) = Jumlah bobot path terkecil dari v1 ke vj
5.  Secara formal, algoritma Dijkstra untuk mencari path terpendek adalah
6.  sebagai berikut:
    1. L = { };
7.  V = {v2, v3,..,vn}.
8.  2. Untuk i = 2, ..., n, lakukan D(i) – w(1, i)
9.  3. Selama v
    n ∉ L lakukan:
10. a. Pilih titik vk ∈ V – L dengan D(k) terkecil.
11. L = L ∪ {vk}
12. b. Untuk setiap vj ∈ V – L lakukan:
    Jika D(j) > D(k) + W(k,j) maka ganti D(j) dengan D(k) + W(k,j)
13. 4. Untuk setiap vj ∈ V, w*(1, j) = D(j)
14. Menurut algoritma di atas, path terpendek dari titik v1 ke vn adalah
15. melalui titik-titik dalam L secara
    berurutan, dan jumlah bobot path terkecilnya adalah D(n).
16. Algoritma Dijkstra dinyatakan dalam pseudo-code berikut ini( Rinaldi
17. Munir 2005 , hal. 414):
    procedure Dijkstra (input m:matriks, a:simpul awal)
18. (
19. Mencari lintasan terpendek dari simpul awal a ke semua simpul lainnya
20. Masukan : matriks ketetanggaan (m) dari graf berbobot G dan simpul
    awal a
21. Keluaran : lintasan terpendek dari a ke semua simpul lainnya
22. )
    Deklarasi
23. s1, s2, ..., sn :integer (tabel integer)
24. d1, d2, ..., dn :integer (tabel integer)
25. i, j, k: integer
    Algoritma
26. ( langkah 0 (Inisialisasi:)
27. for i ← 1 to n do
28. si← 0
    di ←mai
29. endfor
30.
```

(Fitria, 2013).

2.2.4 Arduino Uno

Arduino memiliki beberapa jenis terutama pada arduino *board*. Arduino *board* satu dengan arduino *board* yang lainnya memiliki jenis dan

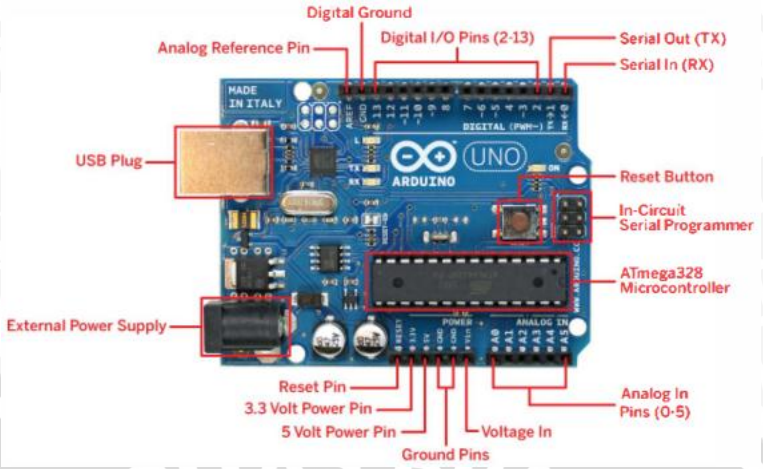
fungsi yang berbeda satu dengan yang lainnya. Jenis – jenis arduino *board* yaitu Arduino Uno, Arduino Mega ADK, Arduino Mini, Arduino Pro, Arduino Pro Mini dan masih banyak lagi. Dalam penelitian ini menggunakan arduino uno yang merupakan salah satu jenis dari arduino *board* yang ada. Arduino uno memiki spesfikasi dan baigian – bagian yang berbeda dari yang lain.

Tabel 2. 1 Spesifikasi Arduino Uno

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM *output*)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

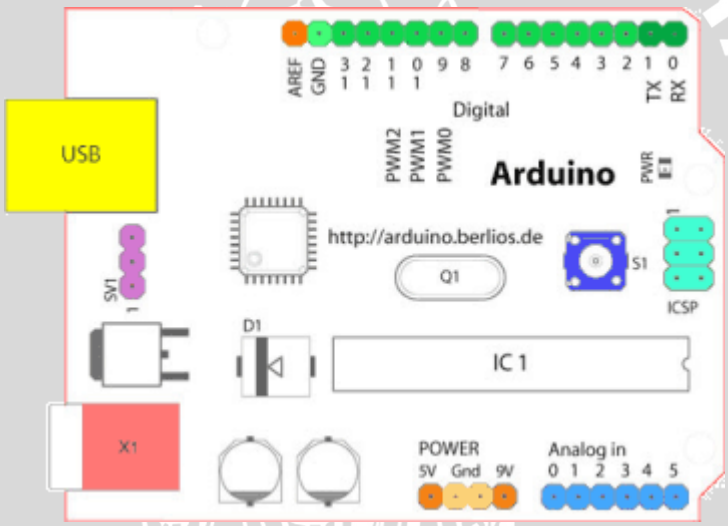
Sumber : (Katia, 2015)

Dibawah ini akan menjelaskan bagian – bagian pada arduino uno secara mendetail.



Gambar 2. 3 Arduino Uno

Sumber : (Badamasi, 2014)



Gambar 2. 4 Bagian Arduino Uno board

Sumber : (Djuandi, 2011)

Pada penelitian ini akan menggunakan arduino uni sebagai mikrokontrolernya, bagian – bagian yang dijelaskan sebagai berikut :

1. 14 pin *input / output* digital (0 – 13)

   Berfungsi sebagai *input* dan *output* yang dapat diatur oleh program. Untuk 6 buah pin 3, 5, 6, 9, 10 dan 11 dapat juga berfungsi sebagai pin analog *output* dimana tegangan *output*nya dapat diatur. Nilai sebuah pin *output*

analog dapat diprogram antara 0 – 255, dimana hal tersebut mewakili nilai tegangan 0 – 5v.

2. USB

   USB disini memiliki beberapa fungsi, deperti halnya memuat program dari komputer ke dalam papan, komunikasi serial antara papan dan komputer dan memberi daya listrik kepada papan

3. Sambungan SV1

   Sambungan atau jumper untuk memilih sumber daya papan, pemilihan dari sumber eksternal atau menggunakan USB.

4. Q1 – Kristal (*quartz crystal oscillator*)

   Kristal adalah sebuah jantung pada arduino uno ini, hal tesebut dikarenakan komponen ini menghasilkan detak – detak yang dikirim ke mikrokontroller agar melakukan sebuah operasi untuk setiap etaknya. Kristal ini dipilih yang berdetak 16juta kali perdetik (16MHz).

5. Tombol *Reset* S1

   Sebagai tombol reset, fungsi dari bagian ini yaitu me-reset arduino *board* agar memuali dari program awal.

6. *In-Circuit Serial Programming (ICSP)*

   Port ICSP memiliki fungsi untuk memprogram mikrokontroller secara langsung tanpa melalui *bootloader*.

7. IC 1 – *Microcontroller* Atmega

   Komponen utama yang tidak dapat dipisahkan dari arduino *board* yang didalamnya mencakup CPU, ROM, dan RAM.

8. X1

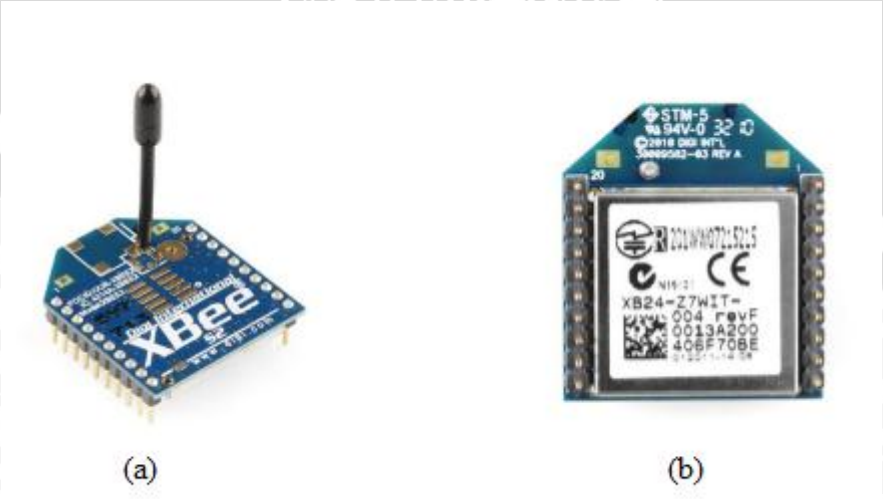
   X1 merupaka sumber daya *eksternal* , apabila arduino dihendaki mendapat daya *eksternal* maka arduino *board* akan diberikan tegangan DC antara 9 – 12V.

9. 6 pin *input* analog (0 – 5)

   6 pin yang sangat di perlukan untuk pembacaan tegangan yang dihasilkan oleh sebuah sensor analog. Program dapat membaca nilai sebuah *input* antara 0 – 1023 yang dimana teah mewakili nilai tegangan 0 – 5V (Djuandi, 2011).

2.2.5 Modul Wireless Radio Frequency 2,4 GHz

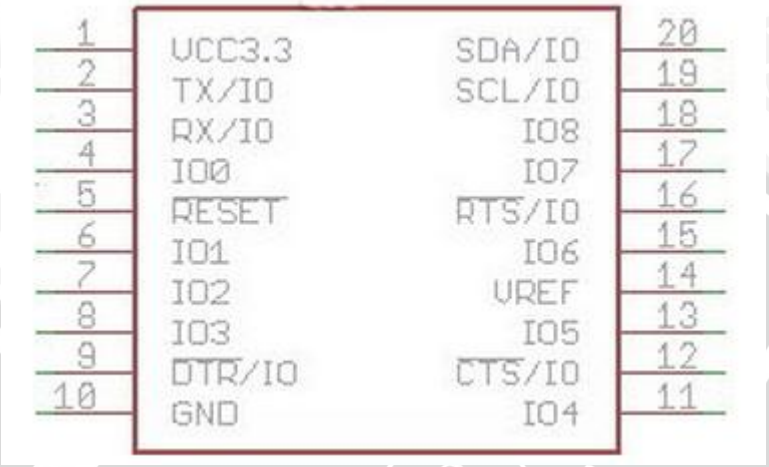
Zigbee adalah sebuah perangkat *wireless transceiver* menggunakan frekuensi radio yang berfungsi sebagai komunikasi 2 arah. Zigbee sendiri merupakan bentuk implementasi dari protokol komunikasi dengan standart IEEE 802.15.4. Dari berbagai modul komunikasi zigbee yang dijumpai di pasaran salah satunya adalah xbee (Hadiyoso, Rizal, Aulia, & Sofie). Xbee ini merupakan salah satu produk dari Digi Internasional. Xbee merupakan sebuah modul RF transceiver menggunakan standart protokol zigbee dan berkerja dalam jangkauan frekuensi 2,4 GHz. Dalam segi antarmuka komunikasi xbee menggunakan komunikasi serial UART (*Universal Asynchronous Receiver Transmitter*), dengan digunakannya antar muka komunikasi UART maka lebih mudah diintegrasikan dengan mikrokontroller. Modul tersebut memberi solusi konektivitas *end-point nirkabel* untuk perangkat *embedded*. Dengan menggunakannya protokol jaringan IEE 802.15.4 mendukung *point-to-multipoint* atau jaringan *peer-to-peer*. Modul Xbee ini ideal untuk aplikasi berdaya rendah dan murah. Pada penelitian ini digunakan modul Xbee series 2 wire antena yang dikonfigurasi untuk komunikasi point to point. XBee tersebut memiliki daya pancar 2 dBm dengan sensitivitas minimum -96 dBm. Untuk spesifikasi lebih lengkap dapat dilihat pada tabel 2.2



Gambar 2. 5 tampilan xbee

(a) xbee tampak depan, (b) xbee tampak belakang

Sumber : http://www.digi.com/



Gambar 2. 6 Pin Xbee S2

Sumber : http://www.seeedstudio.com/

Tabel 2. 2 Spesifikasi Xbee Series 2

Platform	***Xbee Series 2***
Performance	
RF Data Rate	*250 Kbps*
Indoor / Urban Range	*133 ft (40m)*
Outdoor / RF Line-of-Sight Range	*400 ft (120m)*
Transmit Power	*1,25 mW (+1 dBm) / 2mW (+3dBm) boost Mode*
Features	
Adjustable Power	*Yes*
I/O Interface	*3,3V CMOS UART, ADC, DIO*
Frequency Band	*2,4 GHz*
Interference Immunity	*DSSS (Direct Sequence Spread Spectrum)*
Serial Data Rate	*1200 bps – 1 Mbps*
ADC Inputs	*(4) 10-bit ADC inputs*
Digital I/O	*10*

Network & Security	
Encryption	*128-bit AES*
Reliable Packet Delivery	*Retries / Acknowledgments*
IDs and channels	*PAN ID, 64-bit IEEE MAC, 16 channels*
Power Requirements	
Supply Voltage	*2,1 – 3,6 VDC*
Transmit Current	*35 mA / 45 mA boost mode @ 3,3VDC*
Receive Current	*38 mA / 40 mA boost mode @ 3,3VDC*

2.2.6 RFID Reader RDM6300

RDM6300 125KHz *cardreader mini-module* adalah perangkat yang dirancang untuk membaca kode dari kartu 125KHz kompatibel read-only tag dan membaca / menulis kartu. Hal ini dapat diterapkan dalam keamanan kantor / rumah, identifikasi pribadi, kontrol akses, anti-pemalsuan, sistem kontrol mainan dan produksi interaktif dll. RDM6300 merupakan pengembangan dari versi sebelumnya, RDM630 dimana modul RDM6300 memiliki sensitivitas dan jarak baca yang lebih jauh.

Gambar 2. 7 Tampilan RDM6300

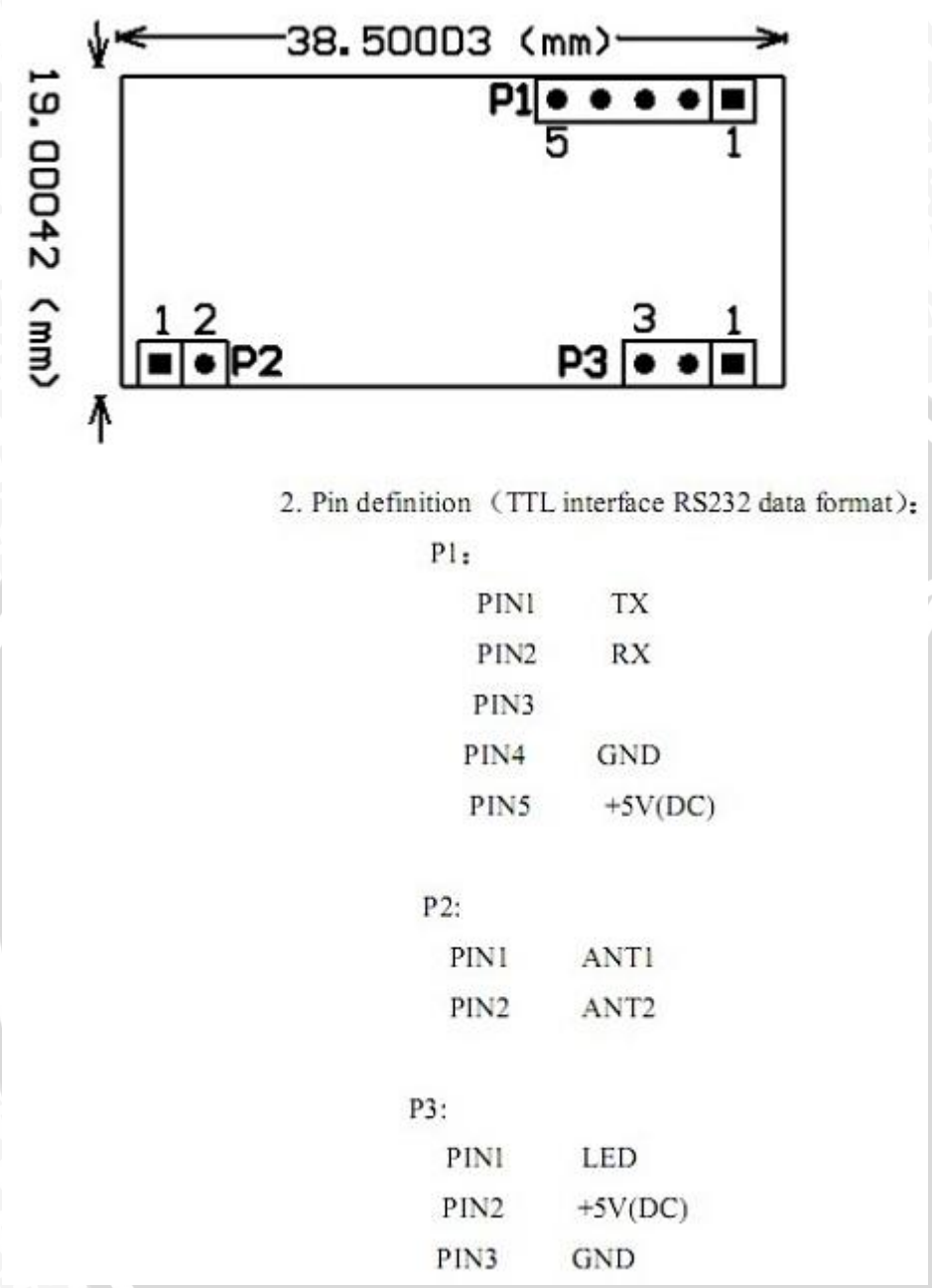
(Sumber: Sparkfun.com)

RDM6300 mendukung antena eksternal dengan jarak baca efektif maksimal 50 mm. Fitur RDM6300 dilengkapi dengan kecepatan decoding kurang dari 100 ms. Sebagai support tatap muka pengguna, RDM6300 mendukung

interface UART. RDM6300 digunakan untuk RFID tipe EM4100 dengan jangkauan 125 KHz. Sebagai pelengkap, modul dilengkapi dengan tambahan led eksternal dan *driver* buzzer.

Tabel 2. 3 Spesifikasi modul RDM6300

Tipe Modul	**RFID**
Weight	15.00g
Model	IM120618002
Board Size	3.8 x 1.8 x 1.2cm
Version	1
Operation Level	Digital 5V
Power Supply	External 5V



2. Pin definition (TTL interface RS232 data format):

P1:

PIN1	TX
PIN2	RX
PIN3	
PIN4	GND
PIN5	+5V(DC)

P2:

PIN1	ANT1
PIN2	ANT2

P3:

PIN1	LED
PIN2	+5V(DC)
PIN3	GND

Gambar 2. 8 Dimensi RDM6300

(Sumber : Sparkfun, 2009)

2.2.7 Aplikasi Desktop

Aplikasi berbasis desktop merupakan pengembangan aplikasi perangkat lunak (*software*) yang dijalankan di tiap client. Pada server terdapat data base yang kemudian pada masing-masing klien diinstal sebuah aplikasi. Bahasa pemograman yang digunakan untuk aplikasi tipe ini biasanya adalah Borland

Delphi, Visual Basic, Java Netbean, dsb. Pada aplikasi berbasis desktop, aplikasi dikompilasi pada komputer server dan dapat langsung digunakan.

2.2.8 MySQL

MySQL adalah sebuah sistem manajemen database yang paling popular yang merupakan Open Souce SQL database yang dikembangkan,didistribusikan, dan didukung oleh Oracle Corporation. MySQL merupakan salah satu RDBMS (*Relational Data Base Managemnet System*) terbaik yang digunakan untuk mengembangkan web berbasis aplikasi perangkat lunak.

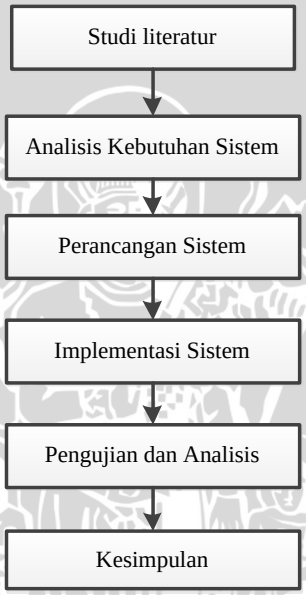
Data Base merupakan kumpulan data yang terstruktur. Misalnya untuk MySQL server, diperlukan sebagai sebuah sistem manajemen Data Base agar dapat dengan mudah menambah, mengakses, serta memproses data yang disimpan dalam computer data base. SQL (*Structured Query Language*) merupakan bahasa standar yang paling umum digunakan dalam mengakses *database*. MySQL bersifat *Open Source*, yang artinya siapa saja dapat menggunakan dan memodifikasinya dengan mengunduh perangkat MySQL secara gratis.

MySQL Server bekerja di client/server atau pada sistem embedded. Data base MySQL merupakan sistem client/server yang terdiri dari server SQL multi-threaded yang mendukung *backends* yang berbeda, beberapa program client dan *libraries* yang berbeda, alat administrasi dan beberapa antarmuka aplikasi pemrograman (*Application Programming Interfaces* atau APIs).

BAB III
METODE PENELITIAN

Dalam perancangan sistem terdapat proses-proses yang saling berkaitan agar menciptakan suatu perancangan yang terstruktur dengan baik. Pada bab ini akan dijelaskan metode yang digunakan dan langkah-langkah yang dilakukan dalam penelitian ini.

Adapun tahapan penelitian dapat digambarkan dalam bentuk diagram blok yang ditunjukkan pada gambar 3.1



Gambar 3. 1 Diagram Blok Metode Penelitian

3.1 Studi Literatur

Pada penelitian ini dibutuhkan studi literatur yang bertujuan untuk mencari dasar-dasar teori yang dapat dijadikan bahan referensi dalam merancang sistem. Materi mengenai *Smart Home*, *Wireless Sensor Network*, dan *Data Base* sebagai dasar penelitian diperoleh dari buku, jurnal, maupun artikel di internet agar dapat dianalisis agar dapat diimplementasikan dan menciptakan sistem yang handal.

3.2 Analisis Kebutuhan Sistem

Analisis kebutuhan dalam merancang sistem tersebut terdiri dari kebutuhan fungsional, kebutuhan data dan perangkat. Kebutuhan fungsional yang terkait dengan sistem antara lain:

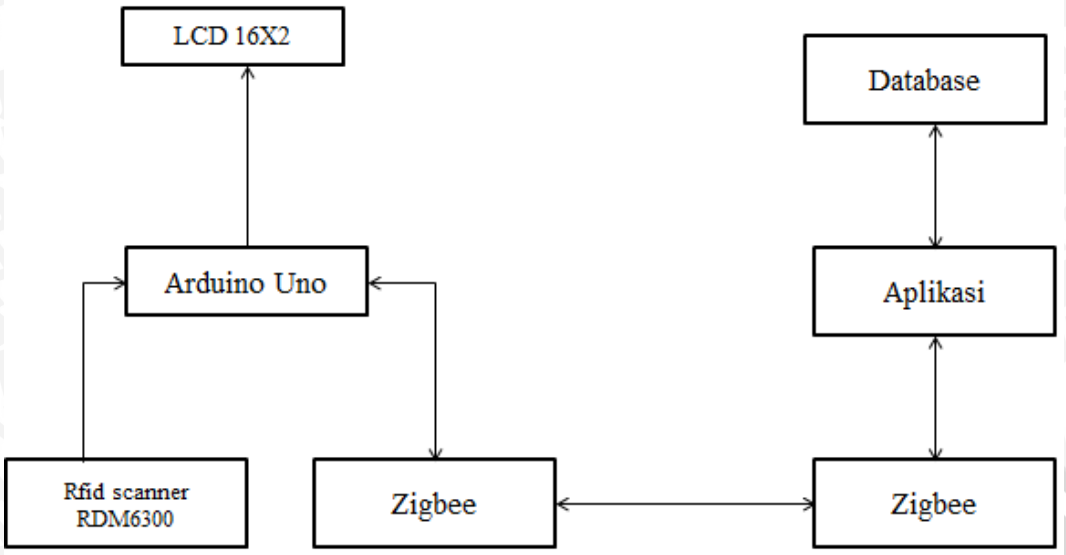
a) Sistem dapat melakukan eksekusi sesuai masukan yang diinginkan pengguna
b) Sistem dapat mengirim perintah ke node/objek setiap saat sesuai dengan keinginan pengguna.
c) Sistem dapat menampilkan data sesuai hasil yang diinginkan berdasarkan rute yang dibuat.

Analisis kebutuhan data dan perangkat diperlukan untuk mengetahui kebutuhan sistem yang terkait dengan pengolahan data, perangkat lunak dan perangkat keras. Kebutuhan data dan perangkat antara lain:

1. Kebutuhan data lapangan, diambil pada pusat perbelanjaan Giant SPM Dinoyo Malang. Kebutuhan data lapangan yang diambil antara lain:
    - Data barang belanja yang diambil berdasarkan data lapangan.
    - Data jarak antar rak sebagai referensi pemetaan lokasi.
    - Data daftar barang belanjaan pengguna sebagai data pengujian akurasi algoritma penentuan rute.
2. Kebutuhan perangkat keras, antara lain:
    - 1 unit komputer sebagai server
    - 1 perangkat Arduino Uno.
    - 2 buah modul Zigbee Pro Series 2.
    - 1 buah RFID Scanner RDM6300
    - 5 buah kartu RFID tag
    - 1 buah LCD 16x2 bit
3. Kebutuhan perangkat lunak, antara lain:
    - MySQL
    - Borland Delphi 2010.
    - Arduino IDE

3.3 Perancangan Sistem

Pada perancangan aplikasi ini perlu diperhatikan perancangan yang tepat agar proses terarah dan terstuktur. Perancangan sistem secara keseluruhan dapat digambarkan melalui diagram blok seperti pada gambar 3.3.



Gambar 3. 2 Diagram blok sistem

Dari diagram blok di atas, masing-masing objek akan terhubung dengan modul zigbee untuk komunikasi data. Data masukan terbagi dua, yaitu data masukan aplikasi dan data masukan modul *wireless*. Data masukan aplikasi merupakan barang belanja pilihan pengguna yang terdiri ID User, ID Produk, nama produk, kategori dan rak. Data barang belanja dari masing-masing pengguna kemudian diolah oleh aplikasi pada *server* untuk mendapat hasil yang diinginkan untuk disimpan pada *database*. Data masukan modul wireless terdiri dari ID User pada tag RFID dan dikirimkan ke komputer server untuk diolah. Pengolahan data dari modul dilakukan di aplikasi berupa verifikasi ID User. Kemudian data hasil yang disimpan pada *database* akan dipanggil oleh aplikasi untuk dikirimkan ke modul *wireless*.

3.4 Implementasi Sistem

Perancangan sistem direalisasikan dalam sebuah implementasi. Implementasi merupakan perancangan program sampai pada hasil akhir berupa saran rute terpendek yang dilalui untuk memperoleh semua daftar barang belanja . Pada implementasi sistem terdapat beberapa tahapan yaitu:

a. Implementasi perangkat lunak. Dalam pembuatan sistem, beberapa perangkat lunak diperlukan untuk mendukung kerja sistem, antara lain Arduino IDE, Borland Delphi 2010, MySQL. Pada implementasi

sistem, perangkat lunak harus dapat menjalankan fungsi-fungsi antara lain:

- Melakukan proses verifikasi pengguna.
- Melakukan proses pesan dan batal pesan barang belanja yang diinginkan.
- Memberikan hasil berupa rute terpendek menuju barang belanja yang dipilih secara keseluruhan.
- Menerima Id *user* dari modul *wireless* dan mengirimkan hasil pengolahan ke modul *wireless.*
- Menerima Id *user* pada RFID tag dan mengirimkan ke *server*.
- Menerima hasil rute terpendek dari aplikasi *server.*

b. Implementasi perangkat keras. Pada tahap ini perangkat keras diimplementasikan sebagai pendukung jalannya kerja sistem. Perangkat keras digunakan dalam proses penerimaan data masukan dan keluaran untuk diolah aplikasi *server*.

Berdasarkan dari implementasi yang disebutkan di atas, diharapkan sistem dapat berjalan sesuai dengan apa yang telah dirancang dan sesuai dengan apa yang diinginkan, antara lain:

1. Mengatur komunikasi serial dan menyambungkan sistem antara Arduino Uno dengan komputer server.
2. Menampilkan data berupa nama, ID, ID produk, nama produk dan rak dari keseluruhan masukan yang diberikan pengguna.
3. Menampilkan hasil berupa rute terpendek dari seluruh masukan yang diberikan.
4. Melakukan manajemen pengguna pada level admin berupa menambahkan dan menghapus pengguna.

3.5 Pengujian dan Analisis

Pengujian pada penelitian ini dilakukan atas beberapa tahapan mulai dari pengujian untuk proses verifikasi *user*, akurasi hasil, hingga komunikasi antara server dan modul *wireless*. Setelah semua sistem diuji secara keseluruhan, dilakukan analisa untuk mengetahui hasil yang akan dilakukan untuk menarik

kesimpulan dari penelitian yang dilakukan. Hasil pengujian dianalisis untuk menentukan apakah sistem sudah berjalan baik sesuai dengan yang diharapkan.

3.6 Kesimpulan

Pada tahap kesimpulan ini akan dilakukan pengambilan kesimpulan dari analisis hasil pengujian dari sistem. Kesimpulan akan memberikan akurasi atas rumusan masalah yang ada.

BAB IV
PERANCANGAN DAN IMPLEMENTASI SISTEM

Bab ini akan membahas mengenai perancangan dari sistem yang akan dibuat dan berdasarkan hasil perancangan tersebut akan dilakukan implementasi sistem yang terdiri dari implementasi perangkat lunak dan perangkat keras.

4.1 Perancangan

Secara umum, sistem yang dibangun terbagi dua yaitu:

1. Perangkat lunak yang digunakan untuk proses masukan barang belanja dan memberikan rute terpendek.
2. Perangkat keras yang digunakan untuk verifikasi upengguna dan menampilkan hasil rute terpendek yang disarankan.

Proses perancangan ini dibagi menjadi 5 tahap, yaiu analisa kebutuhan, perancangan diagram blok sistem, perancangan perangkat lunak, perancangan perangkat keras, dan perancangan uji coba sistem.

4.1.1 Analisa Kebutuhan

Analisis kebutuhan dalam perancangan bertujuan untuk menggambarkan kebutuhan-kebutuhan dari sistem yang terdiri dari kebutuhan fungsional dan kebutuhan non-fungsional. Perangkat lunak dan perangkat keras akan dianalisis sesuai dengan perannya dalam membangun sistem sehingga akan mempermudah dalam mendesain dan mengimplementasikan sistem.

4.1.1.1 Deskripsi Umum Sistem

Secara umum sistem akan menampilkan hasil berupa saran urutan posisi rak yang harus dilalui untuk menghemat waktu belanja. Sistem yang dirancang berupa aplikasi berbasis desktop dan modul *wireless*. Aplikasi desktop dirancang untuk melakukan proses pemilihan barang belanja dan mencari rute terpendek menuju seluruh posisi barang belanja yang akan disimpan dalam database. Modul *wireless* dirancang untuk melakukan pengecekan user dan melakukan *request* rute yang sudah tersimpan di dalam database untuk ditampilkan pada layar LCD 16x2 bit.

Alur kerja dari sistem ini yaitu sebelum belanja pengguna melakukan proses pilih barang belanja yang ingin dibeli, data barang belanjaan akan disimpan didalam database. Saat akan belanja, pengguna melakukan proses cek id menggunakan kartu RFID tag. ID pengguna akan dikirim ke server kemudian server akan mengirim saran rute terpendek ke modul *wireless* sesuai ID pengguna.

4.1.1.2 Kebutuhan Fungsional

Analisis kebutuhan fungsional meliputi beberapa hal, antara lain:

a) Sistem dapat melakukan eksekusi sesuai masukan yang diinginkan pengguna. Saat sistem dijalankan pengguna dapat melakukan operasi belanja berupa pilih barang belanja, hapus barang belanja dan cek daftar belanja yang telah dipilih.

b) Sistem dapat mengirim perintah ke server setiap saat sesuai dengan keinginan pengguna. Saat sistem berjalan pengguna dapat melakukan proses *request* kepada *server* rute menuju posisi barang belanja setiap saat melalui modul *wireless.*

c) Sistem dapat menampilkan data sesuai hasil yang diinginkan berdasarkan algoritma dijkstra.

4.1.1.3 Kebutuhan Data dan Perangkat

Analisis kebutuhan Data dan Perangkat meliputi beberapa hal, antara lain:

1. Kebutuhan data lapangan, diambil pada pusat perbelanjaan Giant SPM Dinoyo Malang. Kebutuhan data lapangan yang diambil antara lain:
   - Data barang belanja yang diambil berdasarkan data lapangan.
   - Data jarak antar rak sebagai referensi *cost*(bobot) pemetaan lokasi.
   - Data daftar barang belanjaan pengguna sebagai data pengujian akurasi algoritma penentuan rute. Untuk memperoleh data masukan yang valid dilakukan penyebaran kuisioner menggunakan aplikasi *google form* pada 30 responden. Kuisioner berisikan mengenai daftar barang belanja apa saja yang dibeli oleh responden saat pergi ke pusat perbelanjaan.
2. Kebutuhan Perangkat Keras
   - 1 unit komputer sebagai server
   - 1 perangkat Arduino Uno.

- 2 modul Zigbee Pro Series 2.
- 1 modul RFID Scanner RDM6300
- 5 kartu RFID tag
- 1 LCD 16x2 bit

3. Kebutuhan Perangkat Lunak
   - MySQL
   - Borland Delphi 2010.
   - Arduino IDE

4.1.2 Diagram Blok Perancangan Sistem

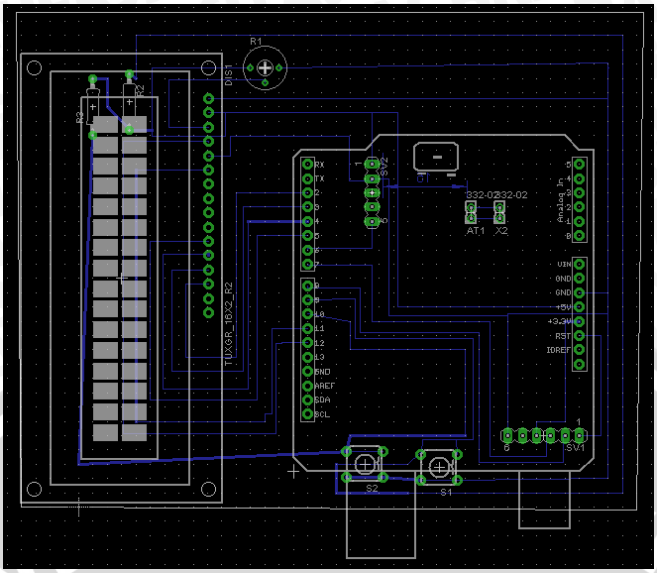
Pada sistem ini perlu diperhatikan perancangan yang tepat agar proses terarah dan tersrtuktur. Perancangan sistem secara keseluruhan dapat digambarkan melalui diagram blok seperti pada Gambar 3.3.

Diagram blok menggambarkan bahwa komunikasi antara modul *wireless* dan *server* dilakukan melalui modul zigbee. Data kemudian diolah pada aplikasi yang terdapat pada komputer server yang kemudian hasilnya dikirimkan kembali ke modul *wireless*.

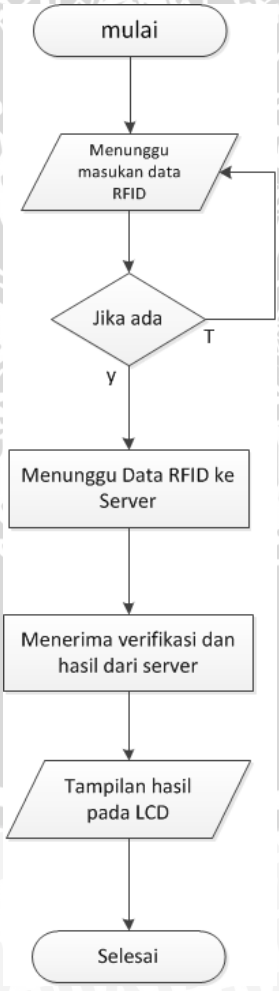
4.1.3 Perancangan Perangkat Keras

Perancangan perangkat keras bertujuan untuk membangun sistem komunikasi di antara semua objek agar dapat menerima dan mengirim data ke *server*.

Perangkat keras pada sistem ini adalah Arduino Uno, RFID reader RDM6300, LCD 16x2 bit dan modul Zigbee. RFID *reader* RDM6300 melakukan proses pembacaan id pada kartu RFID sebagai proses verifikasi *user*. Modul Zigbee berperan sebagai *transmitter* untuk berkomunikasi dengan *server*. Protokol yang digunakan untuk komunikasi adalah Zigbee AT Mode secara *unicast*. Hasil pengiriman dari *server* akan ditampilkan pada LCD 16x2 bit. Desain rangkaian dan diagram alir proses kerja dari perangkat keras dapat dilihat pada Gambar 4.1 dan 4.2.



Gambar 4. 1 Desain Perangkat Keras



Gambar 4. 2 Diagram Alir Proses Kerja

4.1.4 Perancangan Perangkat Lunak

4.1.4.1 Perancangan Paket Data

Perancangan paket data pada penelitian ini terdiri atas dua bagian yaitu paket data antara Arduino ke server dan sebaliknya. Perancangan paket data dari Arduino ke komputer diperlukan dalam proses pengiriman data untuk proses verifikasi pengguna dan menampilkan hasil saran rute yang harus dilewati. Server akan menerima data berupa *RFID Tag Number* dan Arduino akan menerima hasil rute saran dari server.

a. Paket Data dari Arduino ke Komputer

Rancangan paket data dari Arduino ke komputer dapat dilihat pada Tabel 4.1.

Tabel 4. 1 Rancangan Paket Data dari Arduino ke Komputer

Panjang Data	1 Byte	12 Byte	1 Byte
Nama Slot	*	Nilai ID	#

b. Paket Data dari Komputer ke Arduino

Rancangan paket data dari komputer ke Arduino dapat dilihat pada Tabel 4.2.

Tabel 4. 2 Rancangan Paket Data dari Komputer ke Arduino

Panjang Data	1 Byte	2 Byte	1 Byte	n Byte	1 Byte
Nama Slot	*	Rak	-	Nama produk	#

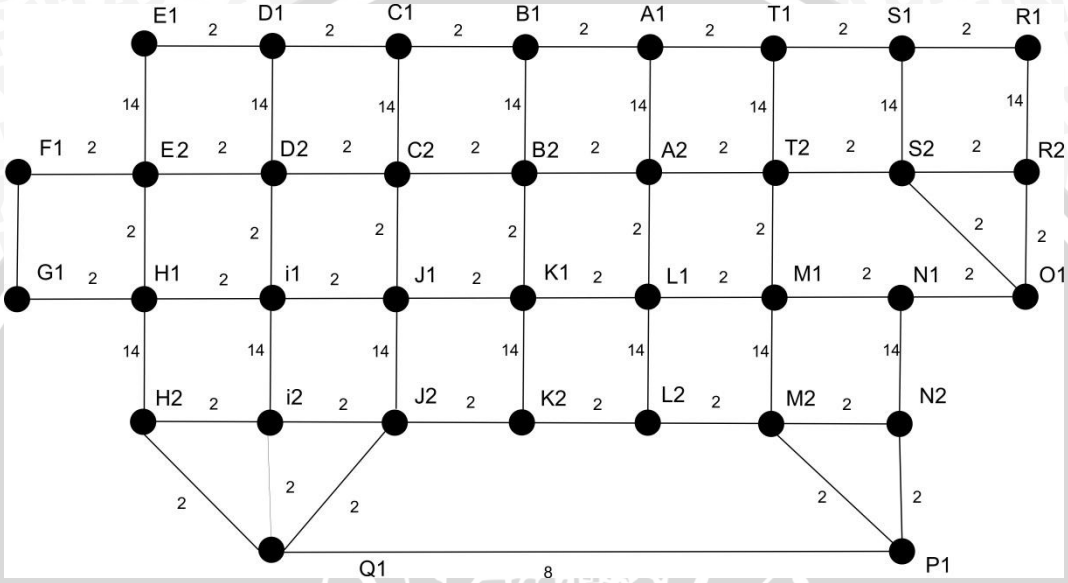
4.1.4.2 Perancangan Desain Pemetaan Lokasi

Pemetaan dilakukan sebagai langkah untuk mengetahui jarak menuju setiap lokasi barang belanja pada daftar belanja yang selanjutnya akan dipakai sebagai *cost* (bobot) dalam graf. Lokasi pemetaan dilakukan pada Giant SPM Dinoyo Malang. Dalam menentukan jarak digunakan parameter rak dimana setiap rak dianggap sebagai *vertex*.

Giant SPM Dinoyo memiliki 20 rak dengan posisi dan jarak yang statis dimana panjang rak terbagi atas rak panjang dan rak pendek. Setiap rak panjang

dibagi menjadi dua *vertex* untuk mengantisipasi pengambilan barang pada bagian ujung rak yang dijelaskan pada gambar 4.3.

Setiap rak diberi indeks nama untuk mempermudah proses pemetaan dan pencarian lokasi. Rak A1 digunakan sebagai *starting point* belanja dimana pada kondisi nyata rak A1 merupakan lokasi terdekat dari pintu masuk lokasi belanja.

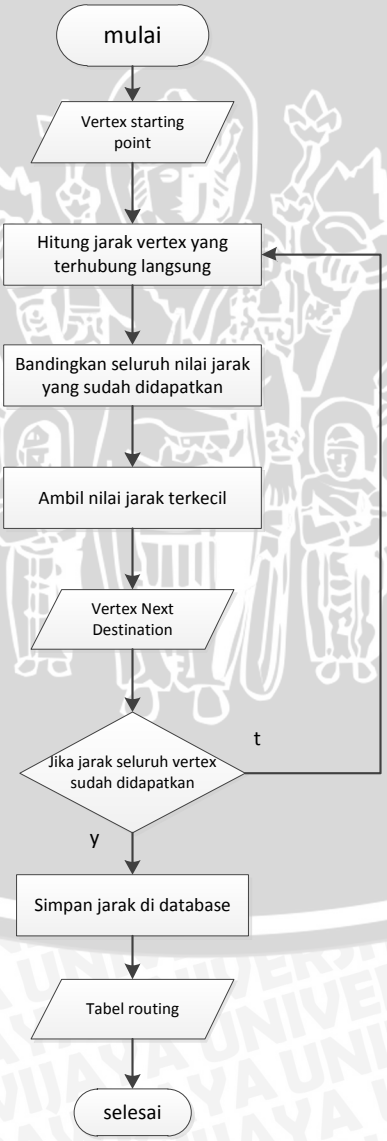


Gambar 4. 3 Desain Pemetaan Lokasi

Untuk menghitung dan menentukan rute dilakukan proses pemetaan jarak satu *vertex* menuju seluruh *vertex* yang ada menggunakan algoritma Dijkstra. Proses pemetaan awal dilakukan dengan menghitung jarak *vertex* yang terhubung langsung dengan *starting point (*SP*)*.

Pada penelitian ini SP pertama adalah A1. *Vertex* yang terhubung langsung dengan A1 adalah A2, B1, dan T1. Jarak antara A1 dengan A2, B2 dan T1 dibandingkan untuk mendapatkan nilai jarak yang terkecil. Saat ditemukan dua atau lebih *vertex* dengan jarak yang sama maka dipilih salah satu *vertex* secara *ascending*. *Vertex* dengan nilai jarak terkecil digunakan sebagai destinasi selanjutnya untuk melakukan proses penghitungan jarak pada *vertex* lain yang terhubung langsung sementara *vertex* yang sebelumnya telah diketahui jaraknya akan disimpan untuk proses pembandingan selanjutnya. *Vertex* yang dihitung selanjutnya adalah B1 dengan *vertex* yang terhubung langsung adalah B2 dan C1.

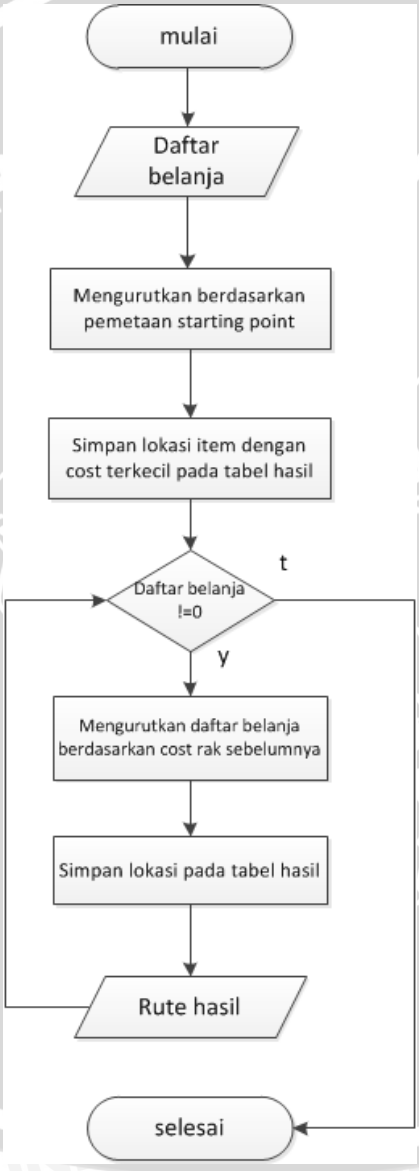
Nilai jarak B2 dan C1 merupakan penjumlahan jarak antara A1 menuju B1 dan B1 menuju *vertex* yang terhubung langsung pada B1. Hasil dibandingkan dengan seluruh *vertex* yang telah diketahui nilainya untuk memilih destinasi selanjutnya. Pada pemetaan ini pembandingan dilakukan antara A2, B2, C1, T1 dan B2 untuk memilih nilai jarak yang terkecil. Proses dilakukan sampai jarak menuju seluruh *vertex* diperoleh. Perhitungan dilakukan secara keseluruhan pada setiap *vertex* sehingga dihasilkan 35 pemetaan. Hasil pemetaan yang sudah dilakukan disimpan pada tabel rute didalam database untuk diakses saat proses pengolahan data pada *server* dalam menentukan rute dimana hasil pemetaan tidak berubah-ubah. Proses pemetaan dijelaskan pada gambar 4.4.



Gambar 4. 4 Proses Pemetaan Jarak

4.1.4.3 Perancangan Algoritma Pengolahan Data Pada Server

Perancangan algoritma pengolahan data pada server mengirim data dimaksudkan untuk memberikan arahan mengenai pembuatan program yang sesuai dengan kerja aplikasi dalam menentukan saran rute terpendek menuju daftar barang belanja.



Gambar 4. 5 Perancangan Pengolahan Data pada Server

Pada aplikasi akan mengolah data masukan berupa barang belanja yang disimpan pada database. Rute lokasi barang belanja pertama akan diurutkan

berdasarkan tabel *routing starting point*, rute yang diambil adalah rute dengan *cost* terendah diantara seluruh daftar belanjaan. Kemudian rute yang dipilih disimpan pada tabel hasil. Rute selanjutnya akan ditentukan berdasarkan tabel routing rak yang sebelumnya dipilih menuju sisa daftar belanja yang belum diproses sampai daftar belanja habis.

4.2 Implementasi

4.2.1 Implementasi Perangkat Lunak

Implementasi perangkat lunak pada penelitian ini merupakan implementasi sistem dari hasil perancangan yang dibuat sebelumnya. Dalam implementasi perangkat lunak ini direpresentasikan dalam bentuk *source code*.

4.2.1.1 Implementasi Algoritma Scan RFID dan Pengiriman data ke Server

Scan RFID berfungsi untuk melakukan verifikasi pengguna ketika akan mengakses modul perangkat keras. Hasil *scan* akan dikirim ke *server* untuk diolah sehingga dapat dikembalikan dalam bentuk hasil sesuai identitas pengguna yang bersangkutan.

```
1.  #include <SoftwareSerial.h>
2.  #include <rdm630.h>
3.  SoftwareSerial xbee(2, 3); // RX, TX
4.  rdm630 rfid(7, 0);
5.  char c,z ;
6.
7.  void setup()  {
8.     Serial.begin(9600);
9.     Serial.println( "Arduino started sending bytes via
    XBee" );
10.    xbee.begin( 9600 );
11.    rfid.begin();
12. }
13.
14. void loop()  {
15. //receive
16.    if (xbee.available() > 0) {
17.     c=xbee.read();
        Serial.write(c);
```

```
18.     }
19.
20.      byte data[10];
21.      byte length;
22.          if(rfid.available()>0){
23.          rfid.getData(data,length);
24.          Serial.println("Data valid");
25.          for(int i=0;i<length;i++){
26.             data[i];
27.          }
28.          for(int i=0; i<length;i++){
29.           Serial.print(data[i],HEX);
30.           Serial.write(' ');
31.           xbee.write(data[i]);
32.          }
33.          Serial.print('#');
34.          xbee.print('#');
             Serial.println();
      }
}
```

Gambar 4. 6 *Source Code* program *scan RFID*

Pada *source code* program di atas memiliki keterangan yaitu:

a) Baris 1-5 merupakan proses deklarasi library dan port yang digunakan.
b) Baris 15-20 merupakan program untuk mengirimkan data melalui zigbee
c) Baris 21-34 merupakan program untuk membaca data id yang masuk kemudian dikonversi dalam bentuk hex.

4.2.1.2 Implementasi Algoritma Pemilihan Barang Belanja

Pemilihan barang belanja dilakukan sebagai masukan rute yang akan ditentukan pada form selanjutnya. Data masukan terdiri dari *id_user*, *nama_produk, id_produk, rak* dan *cost*. Hasil masukan akan disimpan dalam tabel keranjang. *Source Code* dari implementasi algoritma dapat dilihat pada Gambar 4.7.

```
1.  procedure TForm2.kategoriChange(Sender: TObject);
2.  begin
3.        NextGrid1.ClearRows;
4.        dtmodul.ExecSQL('SELECT PRODUK.ID_PRODUK,
5.  PRODUK.NAMA_PRODUK, PRODUK.KATEGORI, PRODUK.RAK,
6.  PRODUK.MERK ,Routing.cost FROM PRODUK, routing where
7.  PRODUK.Kategori = "%s" and PRODUK.rak = routing.rak',
8.                              [kategori.Text],
9.                            dtModul.sqlQuery1);
10.       dtModul.sqlQuery1.First;
11.       ibaris:= 0;
12.       while not dtModul.sqlQuery1.Eof do
13.       begin
14.             NextGrid1.AddRow(1);
15.             NextGrid1.Cell[0, ibaris].AsString :=
16. dtModul.sqlQuery1.Fields[0].AsString;
17.             NextGrid1.Cell[1, ibaris].AsString :=
18. dtModul.sqlQuery1.Fields[1].AsString;
19.             NextGrid1.Cell[2, ibaris].AsString :=
20. dtModul.sqlQuery1.Fields[2].AsString;
21.             NextGrid1.Cell[3, ibaris].AsString :=
22. dtModul.sqlQuery1.Fields[3].AsString;
23.             NextGrid1.Cell[4, ibaris].AsString :=
24. dtModul.sqlQuery1.Fields[4].AsString;
25.             NextGrid1.Cell[5, ibaris].AsString :=
26. dtModul.sqlQuery1.Fields[5].AsString;
27.             inc(ibaris);
28.             dtModul.sqlQuery1.Next;
29.       end;
30. end;
31.
32. // proses pesan sesuai id
33. begin
34.     dtModul.ExecSQL(' Insert into keranjang(id_user,
35. id_produk, nama_produk, rak, cost) VALUES
36. ("%s","%d","%s","%s","%d")',
37.                             [id, idp, nmP, rak, cs],
38.                             dtModul.sqlQuery1);
```

```
34.       ShowMessage('Input sukses');
35.  end;
36.  //form kategori
37.  procedure TForm2.FormActivate(Sender: TObject);
38.  begin
39.       dtmodul.ExecSQL('select `kategori` from `produk`
     group by `kategori` having count(*)>=1',
40.                            [],
41.                          dtModul.sqlQuery1);
42.       while not dtModul.sqlQuery1.Eof do
43.       begin
44.
45.  kategori.Items.Add(dtModul.sqlQuery1.Fields[0].AsStrin
     g);
46.            dtModul.sqlQuery1.Next;
47.       end;
48.
49.
```

Gambar 4. 7 *Source Code* program proses pemilihan barang belanja

Pada gambar di atas memiliki keterangan sebagai berikut:

a) Baris 1-30 merupakan program untuk menampilkan daftar belanja sesuai kolom kategori.
b) Baris 32-40 merupakan program untuk melakukan proses pesan barang belanja.
c) Baris 41-54 merupakan program untuk menampilkan daftar kategori yang tersedia.

4.2.1.3 Implementasi Algoritma Penentuan Rute

Pada implementasi algoritma penentuan rute ditampilkan daftar belanja yang sudah dipilih. Sebelum menentukan rute, pengguna bisa menghapus data belanja yang tidak diinginkan. *Source Code* dari implementasi algoritma dapat dilihat pada Gambar 4.8.

```
1.   procedure TForm3.Belanja_lagiClick(Sender: TObject);
     begin
2.       //start input pertama
```

```
3.      dtModul.ExecSQL('delete from hasil where
4.  id_user="%s" ',
5.      [idU],
6.      dtModul.sqlQuery1);
7.     //insert hasil pertama
8.      dtModul.ExecSQL('insert into hasil
9.  (id_user,nama_produk,rak,cost,id_produk) select
10. id_user,nama_produk,rak,cost, id_produk from dup_ker
11. where dup_ker.rak="%s" and dup_ker.id_user="%s"',
12.     [rck,idU],
13.     dtModul.sqlQuery2);
14.     //del field sebelumnya
15.      dtModul.ExecSQL('delete from dup_ker where
16. id_user="%s" and rak="%s" ',
17.     [idU,rck],
18.     dtModul.sqlQuery3);
19.     //ShowMessage(label3.Caption);
20.
21.     dtModul.ExecSQL('SELECT COUNT(DISTINCT Rak) FROM
22. dup_ker WHERE id_user= "%s" ',
23.     [idU],
24.     dtModul.sqlQuery4);
25.
26.     byk := dtModul.sqlQuery4.Fields[0].AsInteger;
27.     for I := byk downto 1 do
28.     begin
29.       // ambil next rak
30.     dtModul.ExecSQL('select rak from hasil where
31. hasil.no=(select MAX(hasil.no) from hasil) and
32. (id_user="%s") ',
33.     [idU],
34.     dtModul.sqlQuery5);
35.     nDest := dtModul.sqlQuery5.Fields[0].AsString;
36.     dtModul.sqlQuery5.Fields[0].AsString;
//tampilkan rute selanjutnya
    dtModul.ExecSQL('select
dup_ker.id_user,dup_ker.nama_produk,dup_ker.rak,%s.cos
t,dup_ker.id_produk from dup_ker, %s where
```

```
dup_ker.rak=%s.ke and dup_ker.id_user="%s" order
by %s.cost asc',
    [nDest,nDest,nDest,idU,nDest],
    dtModul.sqlQuery6);
    baris:=0;
      while not dtModul.sqlQuery6.Eof do
      begin
      NextGrid2.AddRow(1);
            NextGrid2.Cell[0, baris].AsString :=
dtModul.sqlQuery6.Fields[0].AsString;
            NextGrid2.Cell[1, baris].AsString :=
dtModul.sqlQuery6.Fields[1].AsString;
            NextGrid2.Cell[2, baris].AsString :=
dtModul.sqlQuery6.Fields[2].AsString;
            NextGrid2.Cell[3, baris].AsInteger :=
dtModul.sqlQuery6.Fields[3].AsInteger;
            NextGrid2.Cell[4, baris].AsString :=
dtModul.sqlQuery6.Fields[4].AsString;
            inc(baris);
            dtModul.sqlQuery6.Next;
       end;
  //tempel
       idUs := NextGrid2.Cell[0,0].AsString;
       nmp := NextGrid2.Cell[1,0].AsString ;
       rak := NextGrid2.Cell[2,0].AsString ;
       cst := NextGrid2.Cell[3,0].AsInteger;
       idPr:= NextGrid2.Cell[4,0].AsString;

  //insert tempelan
    dtModul.ExecSQL('insert into
hasil(id_user,nama_produk,rak,cost,id_produk) values
("%s", "%s", "%s", "%d", "%s") ',
    [idU, nmp, rak, cst, idPr],
    dtModul.sqlQuery7);
    NextGrid2.ClearRows;
  //hapus dari duplikat
    dtModul.ExecSQL('delete from dup_ker where
id_user="%s" and rak="%s" ',
```

```
83.      [idU,rak],
84.    dtModul.sqlQuery8);
85.
86.     end;
87.     ShowMessage('berhasil');
88.     end;
89.
90.  //operasi show keranjang belanja dan hapus
91.  procedure TForm3.FormActivate(Sender: TObject);
92.     dtmodul.ExecSQL('Select Id_user, Id_produk,
93.  Nama_produk, Rak, Cost from keranjang where Id_user =
94.  "%s" ',
95.                     [idUser],
96.                     dtModul.sqlQuery2);
97.      dtModul.sqlQuery2.First;
98.      line:= 0;
99.
100. procedure TForm3.Hapus1Click(Sender: TObject);
101. begin
102.      Baris := NextGrid3.SelectedRow;
103.      idU := NextGrid3.Cell[0, Baris].AsString;
104.      idp := NextGrid3.Cell[1, Baris].AsInteger;
105.
106.      dtModul.ExecSQL('DELETE FROM Keranjang WHERE
107. id_produk = %d and id_user = "%s"',
108.                      [idp, idU],
109.                      dtModul.sqlQuery1);
110.
111.      ShowMessage('Delete sukses');
112. end;
```

Gambar 4. 8 *Source Code* Penentuan Rute

Pada Gambar 4.8 Source Code di atas memiliki keterangan yaitu:

1) Baris 1-104 merupakan program untuk menentukan urutan rute terpendek menuju seluruh lokasi barang belanja.
2) Baris 106-114 merupakan program untuk menampilkan daftar barang belanja yang sudah dipilih sebelumnya.

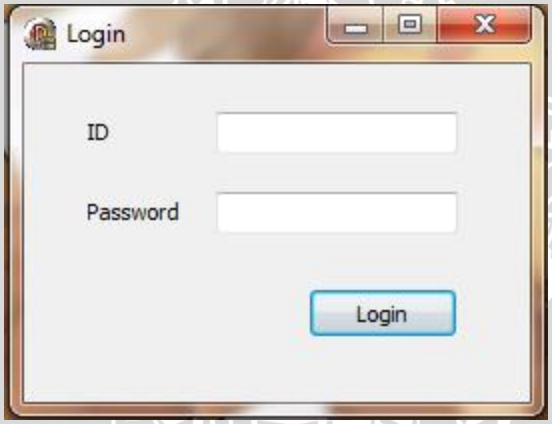
3) Baris 116-128 merupakan program untuk menghapus barang belanja yang sudah dipilih.

4.2.2 Implementasi Antar Muka

Implementasi antarmuka perangkat lunak pada sistem untuk dibuat dengan berbasis *desktop application*.

4.2.2.1 Halaman Log In

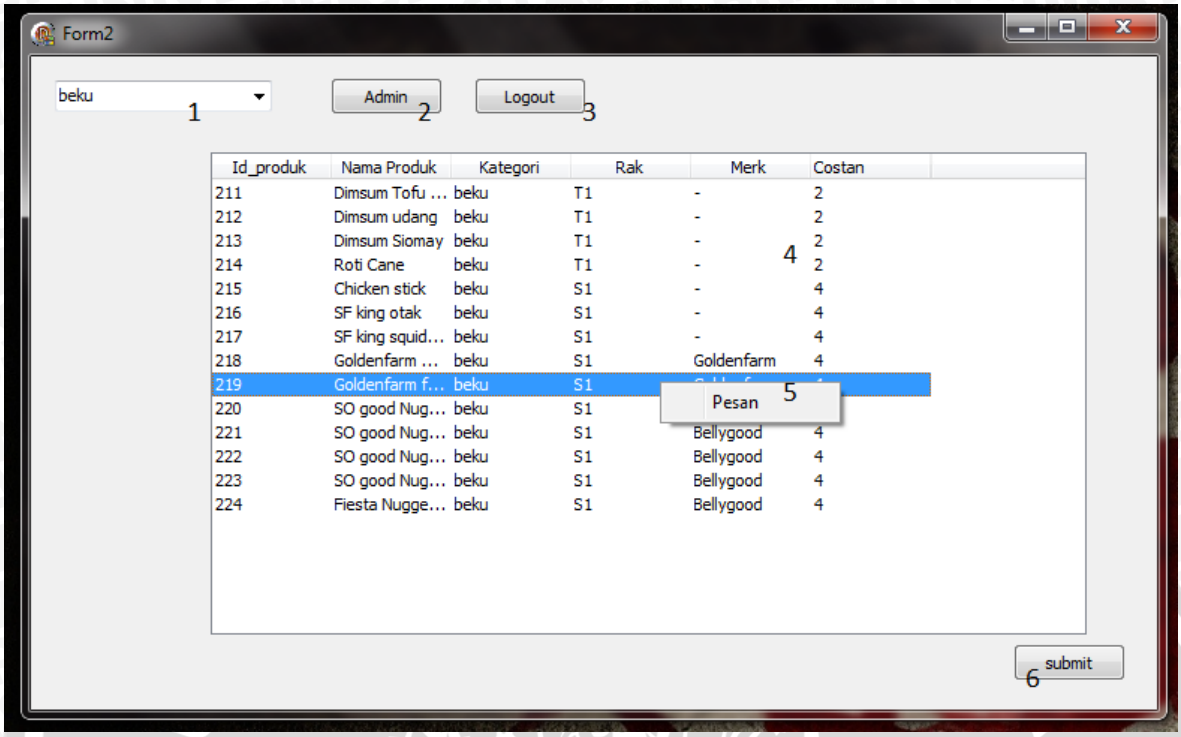
Halaman *login* merupakan antarmuka pengguna sebagai akses masuk pengguna dalam menggunakan sistem sesuai dengan hak aksesnya. Hak akses pengguna terdiri dari *admin* dan *guest*. Admin dapat mengakses sekaligus mengubah data *guest* sedangkan *guest* hanya dapat mengakses sistem tanpa dapat mengubahnya. Gambar 4.9 menunjukkan perancangan tampilan *interface* halaman *login*.



Gambar 4. 9 Tampilan Halaman *Login*

4.2.2.2 Halaman Utama

Pada halaman ini berisi fitur menambahkan barang belanja. Untuk mempermudah pencarian diberikan kolom kategori pada sisi kiri halaman. Gambar 4.10 menunjukkan perancangan halaman utama.



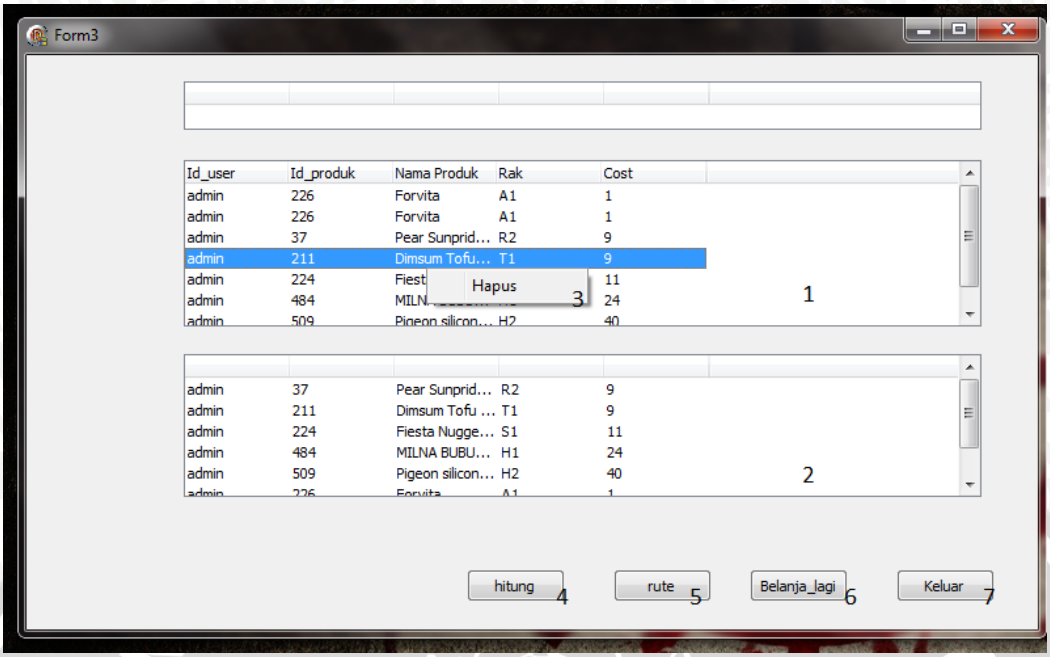
Gambar 4. 10 Tampilan halaman utama

Gambar 4.10 memiliki keterangan sebagai berikut:

1) Kolom kategori untuk menampilkan kategori barang belanja yang tersedia
2) Tombol untuk masuk halaman admin.
3) Tombol untuk keluar dari akun.
4) Tampilan daftar barang belanja sesuai kategori yang dipilih.
5) Proses pesan barang belanja.
6) Tombol submit barang belanja untuk menampilkan hasil seluruh barang belanja.

4.2.2.3 Halaman Tampilan Keranjang Belanja

Pada tahap ini merupakan tampilan daftar pilihan barang belanja pada halaman sebelumnya. pemilihan objek pada tahap sebelumnya. Pada halaman ini pengguna bisa menghapus barang belanja, melakukan proses penentuan rute, melihat hasil rute fix atau belanja kmbali. Gambar 4.12 akan menunjukkan perancangan tampilan halaman Keranjang Belanja.



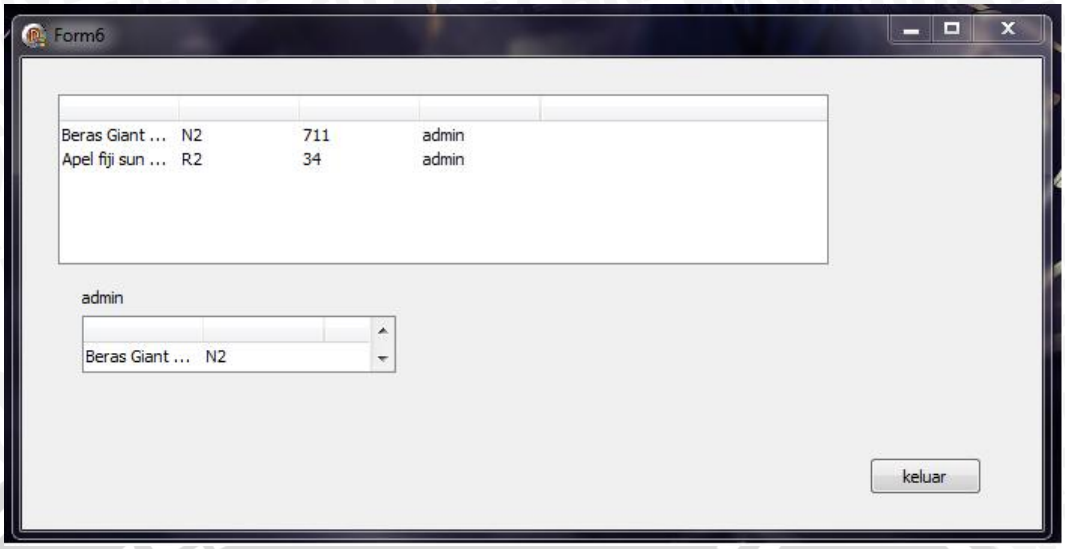
Gambar 4. 11 Halaman keranjang belanja

Gambar 4.11 memiliki keterangan sebagai berikut:

1) Tampilan rute berdasarkan *starting point*.
2) Tampilan daftar belanja secara keseluruhan.
3) Menu pop up untuk menghapus barang belanja.
4) Tombol untuk menentukan rute terpendek.
5) Tombol untuk melihat hasil rute terpendek.
6) Tombol untuk menuju ke halaman utama.
7) Tombol untuk keluar akun.

4.2.2.4 Halaman Hasil

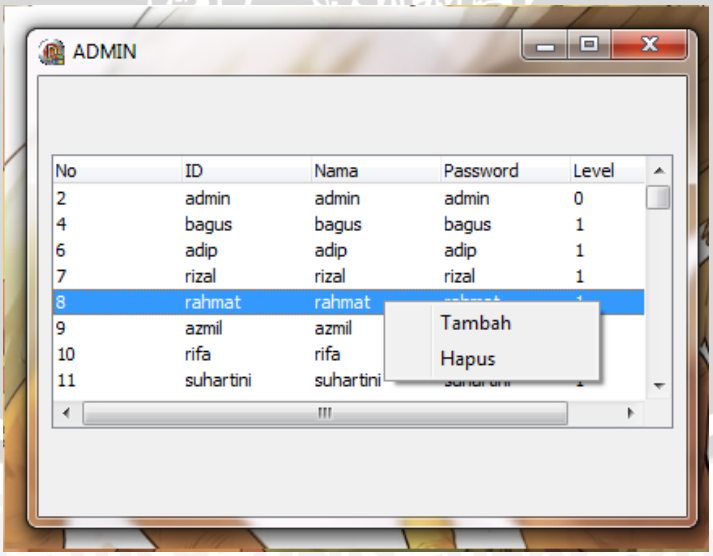
Halaman ini berisikan daftar rute yang disarankan untuk dilewati. Gambar 4.12 akan menunjukkan perancangan tampilan antar muka halaman hasil.



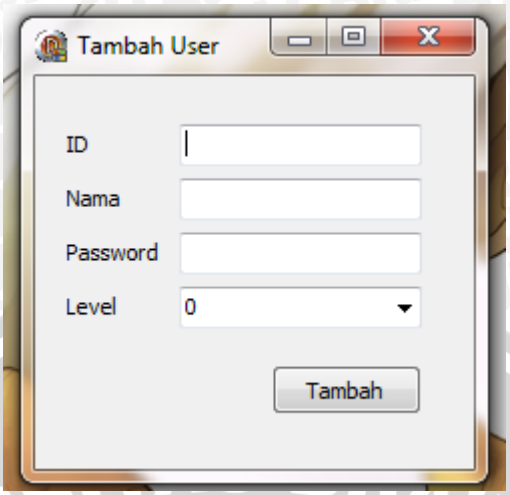
Gambar 4. 12 Tampilan Halaman Hasil Rute Terpendek

4.2.2.5 Halaman Pengaturan Admin

Halaman pengaturan admin merupakan antarmuka pengguna sebagai *administrator* sistem. Admin dapat mengolah data user seperti menambah, menghapus maupun mengubah user dan memberikan hak akses pada masing-masing user. Halaman ini hanya dapat diakses oleh admin. Gambar 4.13 dan 4.14 menunjukkan perancangan tampilan *interface* halaman *admin.*



Gambar 4. 13 Tampilan Halaman awal Pengaturan Admin



Gambar 4. 14 Tampilan Halaman Penambahan user admin

BAB V
PENGUJIAN DAN ANALISIS

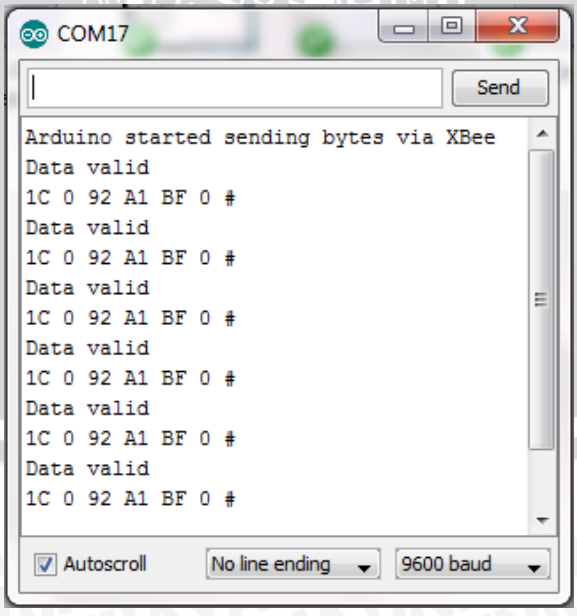
Pada bab ini dilakukan proses pengujian dan analisis dari sistem yang telah dibangun. Tujuan dari dilakukannya pengujian ini adalah untuk mengetahui bahwa semua kebutuhan fungsional maupun non-fungsional yang dirancang sebelumnya telah terpenuhi. Hal-hal yang diuji pada sistem ini yaitu komunikasi data, akurasi rute dan akurasi scan rfid .

Analisis dilakukan untuk mendapatkan kesimpulan dari hasil pengujian. Proses analisis mengacu pada dasar teori sesuai dengan hasil pengujian yang didapatkan pada setiap tahapan pengujian.

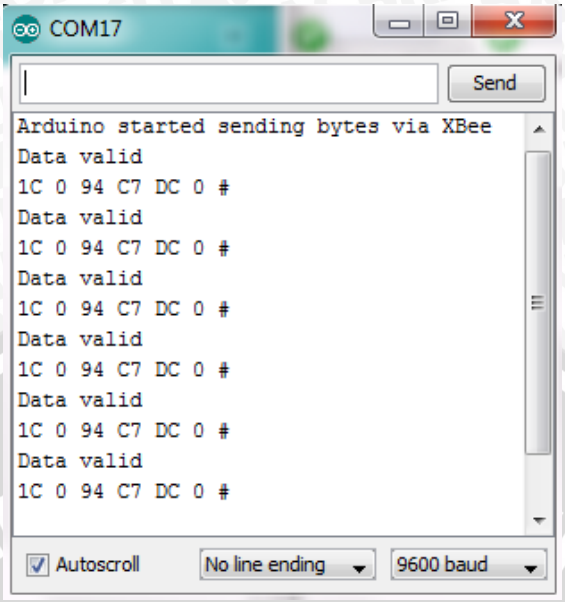
5.1 Pengujian dan Analisis Algoritma RFID Scanner

5.1.1 Pengujian Algoritma RFID Scanner

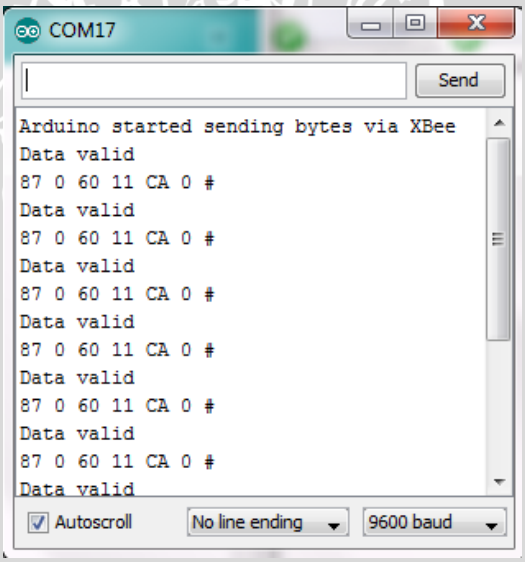
Pengujian algoritma dilakukan untuk melihat apakah algoritma dapat membaca nomor id pada kartu rfid sebagai media verifikasi pengguna. Pengujian dilakukan pada Arduino Uno dengan menggunakan 5 kartu rfid, masing- masing dilakukan sekurang-kurangnya 4 kali pengecekan. Hasil pengujian dapat dilihat pada gambar 5.1.



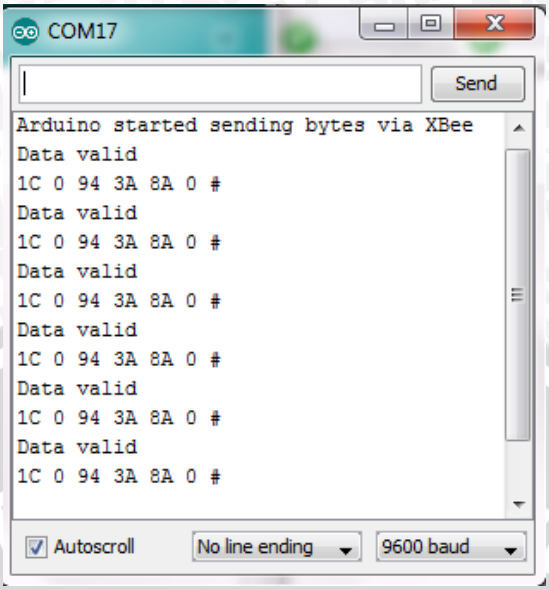
Gambar 5. 1 Pengujian pertama algoritma RFID Scanner



Gambar 5. 2 Pengujian kedua algoritma RFID Scanner



Gambar 5. 3 Pengujian ketiga algoritma RFID Scanner



Gambar 5. 4 Pengujian keempat algoritma RFID Scanner

5.1.2 Analisis Algoritma RFID Scanner

Pengujian Id yang diperoleh akan dikonversi menjadi data HEX kemudian ditampilkan pada serial monitor. Apabila id yang dibaca benar maka akan ditampilkan data valid.

Berdasarkan hasil pengujian, algoritma dapat membaca ID pada kartu dengan baik. Hal ini dibuktikan dengan melakukan konversi data yang ditampilkan menjadi nilai desimal dengn akurasi 100%.

5.2 Pengujian dan Analisis Hasil Rute

5.2.1 Pengujian Hasil Rute

Pengujian rute hasil dilakukan untuk melihat apakah aplikasi memberikan hasil sesuai metode yang dirancang. Rute yang dirancang berupa keputusan pemilihan lokasi selanjutnya. Pemilihan rute selanjutnya dilakukan berdasarkan pertimbangan jarak terpendek menuju rak selanjutnya.

Pengujian algoritma akan menampilkan id_user dan rak. Tujuan dari pengujian tersebut adalah untuk mengetahui tingkat akurasi sistem menggunakan algoritma dijkstra dalam menentukan rute selanjutnya menuju lokasi barang belanja. Hasil rute terpendek ini akan dibandingkan dengan perhitungan rute terpendek secara manual.

Berdasarkan hasil analisis dilakukan aplikasi memberikan hasil rute yang dijelaskan sebagai berikut:

- Aji : rute dilewati 12, rute akurat 12, rute salah 0.
- Azmil : rute dilewati 17, rute akurat 17, rute salah 0.
- Adip : rute dilewati 19, rute akurat 19, rute salah 0.
- Bagus : rute dilewati 16, rute akurat 16, rute salah 0.
- Deby : rute dilewati 25, rute akurat 25, rute salah 0.
- Dhyk : rute dilewati 21, rute akurat 21, rute salah 0.
- Dina : rute dilewati 23, rute akurat 23, rute salah 0.
- Fatih : rute dilewati 18, rute akurat 18, rute salah 0.
- Fitria : rute dilewati 18, rute akurat 18, rute salah 0.
- Firdha : rute dilewati 22, rute akurat 22, rute salah 0.
- Freddy : rute dilewati 17, rute akurat 17, rute salah 0.
- Hariansah : rute dilewati 22, rute akurat 22, rute salah 0.
- Hutomo : rute dilewati 15, rute akurat 15, rute salah 0.
- Ika : rute dilewati 24, rute akurat 24, rute salah 0.
- Jenar : rute dilewati 21, rute akurat 21, rute salah 0.
- Mawar : rute dilewati 25, rute akurat 25, rute salah 0.
- Mes : rute dilewati 13, rute akurat 13, rute salah 0.
- Mikko : rute dilewati 18, rute akurat 18, rute salah 0.
- Mimi : rute dilewati 23, rute akurat 23, rute salah 0.
- Nova : rute dilewati 21, rute akurat 21, rute salah 0.
- Panji : rute dilewati 16, rute akurat 16, rute salah 0.
- Rahmat : rute dilewati 21, rute akurat 21, rute salah 0.
- Rasid : rute dilewati 11, rute akurat 11, rute salah 0.
- Rifa : rute dilewati 18, rute akurat 18, rute salah 0.
- Ririn : rute dilewati 19, rute akurat 19, rute salah 0.
- Rizal : rute dilewati 17, rute akurat 17, rute salah 0.
- Setyan : rute dilewati 19, rute akurat 19, rute salah 0.
- Suhartini : rute dilewati 21, rute akurat 21, rute salah 0.
- Tiyo : rute dilewati 21, rute akurat 21, rute salah 0.
- Ziya : rute dilewati 16, rute akurat 16, rute salah 0.

5.2.2 Analisis Hasil Rute

Berdasarkan hasil pengujian rute yang dirancang dianalisis sistem dapat melakukan penentuan rute dengan baik. Dari 30 pengujian yang dilakukan sistem memberikan rute dengan pertimbangan *cost* terpendek posisi barang selanjutnya sebanyak 576 kali dengan akurasi 100%.

5.2.2.1 Proses Pencarian Rute Terpendek

Pengujian dilakukan untuk menjelaskan pencarian rute optimal pada aplikasi. Proses pencarian dilakukan sebanyak dua kali menggunakan id_user aaa.pada saat pengguna memiliki satu barang belanja dan lebih dari satu barang belanja. Hasil perhitungan kemudian dibandingkan untuk menentukan total *cost.*

- Satu Barang Belanja

  Dalam pengujian ini, diuji saat user aaa memiliki satu pilihan belanja yaitu Milna Bubur Bayi.

  1. Posisi barang belanja adalah rak H1 dengan total *cost* 24. Berdasarkan pemetaan pada SP, vertex dengan cost terkecil adalah B1(cost=2, total cost=2).
  2. Posisi aktual pada vertex B1 dengan vertex terhubung langsung B2 dan C1. Vertex dengan cost terkecil adalah C1 (cost=2, total cost=4).
  3. Posisi aktual pada vertex C1 dengan vertex terhubung langsung C2 dan D1. Vertex dengan cost terkecil adalah D1 (cost=2, total cost=6).
  4. Posisi aktual pada vertex E1 dengan vertex terhubung langsung D2 dan E1. Vertex dengan cost terkecil adalah E1 (cost=2, total cost=8).
  5. Posisi aktual pada vertex E1 dengan vertex terhubung langsung E2 . Vertex dengan cost terkecil adalah E2 (cost=14, total cost=22).
  6. Posisi aktual pada vertex B1 dengan vertex terhubung langsung F1 dan H1. Vertex dengan cost terkecil adalah H1 (cost=2, total cost=24).

Berdasarkan hasil pengujian diketahui untuk mendapatkan total *cost* yang optimal menuju Milna Bubur Bayi pada rak H1, sistem memberikan rute dengan baik sesuai total *cost* terkecil pada SP. Sistem memberikan rute A1-B1-C1-D1-E1-E2-H1 untuk dilalui *user.*

- Banyak Barang Belanja

Dalam pengujian ini, diuji saat user aaa memiliki empat pilihan belanja yaitu Sania 2L pada rak A1, Beras Segitiga Rojolele pada rak T1, Saus Pasta La fonte pada B1 dan Sari Roti Isi Keju pada S1.

1. Berdasarkan pemetaan pada SP posisi barang belanja ada di empat lokasi yaitu A1(0), T1(2), B1(2) dan S1(4). Vertex dengan cost terkecil adalah A1(cost=0, total cost=0).
2. Posisi aktual adalah sania 2L pada rak A1. Barang belanja tersisa adalah Beras Segitiga Rojolele(T1, cost 2), Saus Pasta La Fonte(B1, cost 2) dan Sari Roti Isi Keju(S1, cost 4). Vertex dengan cost terkecil adalah T1(cost=2, total cost=2).
3. Posisi aktual adalah Beras Segitiga Rojolele pada rak T1. Barang belanja tersisa adalah Saus Pasta La Fonte (B1, cost 4) dan Sari Roti Isi Keju(S1, cost 2). Vertex dengan cost terkecil adalah S1(cost=2, total cost=4).
4. Posisi aktual adalah Sari Roti Isi Keju pada rak S1. Barang belanja tersisa adalah Saus Pasta La Fonte (B1, cost 6). Vertex dengan cost terkecil adalah B1(cost=6, total cost=10) melalui T1, A1

Berdasarkan hasil pengujian diketahui sistem memberikan pilihan pemilihan rute belanja dengan baik sesuai total *cost* terkecil pada vertex SP dengan urutan A1, T1, S1, B1.

5.3 Pengujian dan Analisis Akurasi Rute Hasil

5.3.1 Pengujian Akurasi Rute Hasil

Pengujian dilakukan dengan membandingkan hasil rute yang ditampilkan dengan perhitungan secara manual. Hasil perhitungan kemudian dibandingkan untuk menentukan total *cost.* Pengujian algoritma akan menampilkan id_user, nama_produk dan rak. Tujuan dari pengujian tersebut adalah untuk mengetahui tingkat akurasi algoritma dijkstra dalam menentukan total *cost* terendah untuk sistem yang dibangun.

Berdasarkan pengujian tersebut didapatkan hasil seperti yang ditunjukkan sebagai berikut:

- Aji : total *cost* manual 76, total *cost* sistem 76.
- Azmil : total *cost* manual 80, total *cost* sistem 100.
- Adip : total *cost* manual 88, total *cost* sistem 88.
- Bagus : total *cost* manual 86, total *cost* sistem 86 .
- Deby : total *cost* manual 88, total *cost* sistem 88 .
- Dhyk : total *cost* manual 98, total *cost* sistem 98 .
- Dina : total *cost* manual 90, total *cost* sistem 90 .
- Fatih : total *cost* manual 88, total *cost* sistem 90 .
- Fitria : total *cost* manual 78, total *cost* sistem 78 .
- Firdha : total *cost* manual 102, total *cost* sistem 102 .
- Freddy : total *cost* manual 86, total *cost* sistem 86 .
- Hariansah : total *cost* manual 102, total *cost* sistem 102 .
- Hutomo : total *cost* manual 76, total *cost* sistem 76.
- Ika : total *cost* manual 90, total *cost* sistem 96.
- Jenar : total *cost* manual 90, total *cost* sistem 90 .
- Mawar : total *cost* manual 90, total *cost* sistem 90 .
- Mes : total *cost* manual 76, total *cost* sistem 76.
- Mikko : total *cost* manual 80, total *cost* sistem 82 .
- Mimi : total *cost* manual 90, total *cost* sistem 96 .
- Nova : total *cost* manual 90, total *cost* sistem 90 .
- Panji : total *cost* manual 80, total *cost* sistem 80.

- Rahmat : total *cost* manual 90, total *cost* sistem 94 .
- Rasid : total *cost* manual 86, total *cost* sistem 86 .
- Rifa : total *cost* manual 90, total *cost* sistem 90.
- Ririn : total *cost* manual 84, total *cost* sistem 86 .
- Rizal : total *cost* manual 86, total *cost* sistem 86 .
- Setyan : total *cost* manual 100, total *cost* sistem 100.
- Suhartini : total *cost* manual 88, total *cost* sistem 88 .
- Tiyo : total *cost* manual 94, total *cost* sistem 102 .
- Ziya : total *cost* manual 84, total *cost* sistem 86.

5.3.2 Analisis Akurasi Pengujian Rute Hasil

Hasil pengujian menampilkan perbandingan total *cost* yang dilewati antara perhitungan manual dan sistem. Berdasarkan hasil pengujian diketahui aplikasi dapat berjalan sesuai *rute* yang dibuat pada perancangan sistem. Berdasarkan pembandingan total *cost* pada 30 pengujian diketahui bahwa aplikasi memberikan 10 rute yang memiliki total *cost* yang lebih besar daripada perhitungan manual. Hasil ini terjadi saat terdapat beberapa pilihan destinasi lokasi barang dengan nilai *cost* yang sama. Hal ini menyebabkan perbedaan sebagian rute yang dilewati dan berpengaruh terhadap nilai total *cost* yang ditempuh. Proses analisis selengkapnya dilampirkan pada lampiran 1.

5.3.2.1 Proses Pencarian Rute Optimal

Proses ini merupakan penjelasan mengenai proses pencarian rute optimal pada pengujian dan analisis akurasi Rute Hasil. Proses pencarian dilakukan dengan menjumlahkan total *cost* seluruh *vertex* yang dilewati. Pemilihan *vertex* selanjutnya telah dijelaskan pada proses pengolahan data pada server. Saat ditemukan dua atau lebih *vertex* dengan *cost* yang sama maka akan dilakukan simulasi total *cost* pada semua kemungkinan. Id_user yang digunakan sebagai contoh pada penjelasan pengujian ini adalah mes dan ziya. Angka didalam tanda kurung merupakan nilai total *cost vertex*.

- Id_user Mes
  1. Id_user mes memiliki daftar belanja dengan lokasi barang pada A2(14), B1(2), B2(14), C1(4), C2(18), D2(20), E1(8) G1(24), K2(32),

O1(20), Q1(36), R1(6) dan R2(20). *Vertex* dengan nilai terkecil adalah B1 (*cost*=2, total *cost*=2).

2. Pada B1, daftar belanja yang dibandingkan adalah A2(16), B2(14), C1(2), C2(16), D2(18), E1(6), G1(22), K2(30), O1(22), Q1(34), R1(8) dan R2(22). *Vertex* dengan nilai terkecil adalah C1(*cost*=2, total *cost*=4).
3. Pada C1, daftar belanja yang dibandingkan adalah A2(18), B2(16), C2(14), D2(16), E1(4), G1(20), K2(32), O1(24), Q1(32), R1(10) dan R2(24). *Vertex* dengan nilai terkecil adalah E1(*cost*=4, total *cost*=8).
4. Pada E1, daftar belanja yang dibandingkan adalah A2(22), B2(20), C2(18), D2(16), G1(16), O1(26), K2(34), Q1(32), R1(14) dan R2(28). *Vertex* dengan nilai terkecil adalah R1(*cost*=14, total *cost*=22).
5. Pada R1, daftar belanja yang dibandingkan adalah A2(20), B2(22), C2(24), D2(26), G1(30), K2(38), O1(16), Q1(42) dan R2(14). *Vertex* dengan nilai terkecil adalah R2(*cost*=14, total *cost*=36).
6. Pada R2, daftar belanja yang dibandingkan adalah A2(6), B2(8), C2(10), D2(12), G1(16), K2(24) dan Q1(28). *Vertex* dengan nilai terkecil adalah O1(*cost*=2, total *cost*=38).
7. Pada O1, daftar belanja yang dibandingkan adalah A2(6), B2(8), C2(10), D2(12), G1(16), K2(22), dan Q1(28). *Vertex* dengan nilai terkecil adalah A2(*cost*=6, total *cost*=44).
8. Pada A2, daftar belanja yang dibandingkan adalah B2(2), C2(4), D2(6), G1(10), K2(16), dan Q1(26). *Vertex* dengan nilai terkecil adalah B2(*cost*=2, total *cost*=46).
9. Pada B2, daftar belanja yang dibandingkan adalah C2(2), D2(4), G1(8) , K2(14), dan Q1(20). *Vertex* dengan nilai terkecil adalah C2(*cost*=2, total *cost*=48).
10. Pada C2, daftar belanja yang dibandingkan adalah D2(2), G1(6), K2(16), dan Q1(18). *Vertex* dengan nilai terkecil adalah D2(*cost*=2, total *cost*=50).

11. Pada D2, daftar belanja yang dibandingkan adalah G1(4), K2(18), dan Q1(18). *Vertex* dengan nilai terkecil adalah G1(*cost*=4, total *cost*=54).
12. Pada G1, daftar belanja yang dibandingkan adalah Q1(18) dan K2(22). *Vertex* dengan nilai terkecil adalah Q1(*cost*=18, total *cost*=72)
13. Pada G1, daftar belanja yang tersisa adalah K2(4). *Vertex* dengan nilai terkecil adalah K2(*cost*=4, total *cost*=76).

Berdasarkan hasil pengujian sistem dan perhitungan manual memberikan rute lokasi barang belanja dengan optimal pada id_user mes.

- Id_user ziya
  1. Id_user ziya memiliki daftar belanja dengan lokasi barang pada C2(18), D1(6), D2(20), E1(8), F1(23), I2(36), K2(32), L1(16), L2(30), O1(20), P1(34), R1(6), R2(20), S1(4), T2(16). *Vertex* dengan nilai terkecil adalah S1(*cost*=4, total *cost*=4).
  2. Pada S1, daftar belanja yang dibandingkan adalah C2(22), D1(10), D2(24), E1(12), F1(27), I2(40), K2(36), L1(20), L2(34), O1(16), P1(34), R1(2), R2(16), T2(16). *Vertex* dengan nilai terkecil adalah R1(*cost*=2, total *cost*=6).
  3. Pada R1, daftar belanja yang dibandingkan adalah C2(24), D1(12), D2(26), E1(14), F1(29), I2(42), K2(38), L1(22), L2(36), O1(16), P1(34), R2(14), T2(18). *Vertex* dengan nilai terkecil adalah D1(*cost*=12, total *cost*=18).
  4. Pada D1, daftar belanja yang dibandingkan adalah C2(16), D2(14), E1(2), F1(17), I2(30), K2(34), L1(22), L2(36), O1(26), P1(40), R2(26), T2(22). *Vertex* dengan nilai terkecil adalah E1(*cost*=2, total *cost*=20).
  5. Pada E1, daftar belanja yang dibandingkan adalah C2(18), D2(16), F1(15), I2(32), K2(36), L1(24), L2(38), O1(28), P1(40), R2(28), T2(24). *Vertex* dengan nilai terkecil adalah F1(*cost*=15, total *cost*=35).

6. Pada F1, daftar belanja yang dibandingkan adalah C2(5), D2(3), I2(19), K2(23), L1(11), L2(25), O1(15), P1(27), R2(15), T2(11). *Vertex* dengan nilai terkecil adalah D2(*cost*=3, total *cost*=38).
7. Pada D2, daftar belanja yang dibandingkan adalah C2(2), I2(16), K2(20), L1(8), L2(22), O1(12), P1(26), R2(12), T2(8). *Vertex* dengan nilai terkecil adalah C2(*cost*=2, total *cost*=40).
8. Pada C2, daftar belanja yang dibandingkan adalah I2(18), K2(20), L1(6), L2(22), O1(12), P1(26), R2(12), T2(8). *Vertex* dengan nilai terkecil adalah L1(*cost*=6, total *cost*=46).
9. Pada L1, daftar belanja yang dibandingkan adalah I2(20), K2(16), L2(14), O1(6), P1(18), R2(8), T2(4). *Vertex* dengan nilai terkecil adalah T2(*cost*=4, total *cost*=50).
10. Pada T2, daftar belanja yang dibandingkan adalah I2(24), K2(20), L2(18), O1(4), P1(20), R2(4). *Vertex* dengan nilai terkecil adalah O1(4) dan R2(4). Untuk memperoleh hasil paling optimal maka dilakukan perbandingan total *cost* pada kedua kemungkinan yaitu saat *vertex* selanjutnya O1 dan R2.
11. Pembandingan 1
    - Pada O1, daftar belanja yang dibandingkan adalah I2(26), K2(22), L2(20), P1(18), R2(2). *Vertex* dengan nilai terkecil adalah R2(*cost*=2, total *cost*=56) .
    - Pada R2, daftar belanja yang dibandingkan adalah I2(28), K2(24), L2(22), O1(2), P1(20). *Vertex* dengan nilai terkecil adalah O1(*cost*=2, total *cost*=56)
12. Pembandingan 2
    - Pada R2, daftar belanja yang dibandingkan adalah I2(28), K2(24), L2(22), P1(20). *Vertex* dengan nilai terkecil adalah P1(*cost*=20, total *cost*=76) .
    - Pada O1, daftar belanja yang dibandingkan adalah I2(26), K2(22), L2(20), P1(18). *Vertex* dengan nilai terkecil adalah P2(*cost*=18, total *cost*=74) .

13. Pembandingan 3
    - Pada P1 (melewati O1), daftar belanja yang dibandingkan adalah I2(10), K2(6), L2(4) *Vertex* dengan nilai terkecil adalah L2(*cost*=4, total *cost*=80) .
    - Pada P1 (melewati R2), daftar belanja yang dibandingkan adalah I2(10), K2(6), L2(4) *Vertex* dengan nilai terkecil adalah L2(*cost*=4, total *cost*=78).
14. Pembandingan 4
    - Pada L2 (melewati O1), daftar belanja yang dibandingkan adalah I2(6), K2(2). *Vertex* dengan nilai terkecil adalah K2(*cost*=2, total *cost*=82) .
    - Pada L2 (melewati R2), daftar belanja yang dibandingkan adalah I2(6), K2(2). *Vertex* dengan nilai terkecil adalah K2(*cost*=2, total *cost*=80) .
15. Pembandingan 4
    - Pada K2 (melewati O1), daftar belanja selanjutnya adalah I2(4). *Vertex* dengan nilai terkecil adalah I2(*cost*=4, total *cost*=84) .
    - Pada K2 (melewati R2), daftar belanja selanjutnya adalah I2(4). *Vertex* dengan nilai terkecil adalah I2(*cost*=4, total *cost*=82) .

Berdasarkan hasil pengujian pada id_user ziya perhitungan memberikan dua rute lokasi barang belanja yang berbeda. Pemilihan rute yang berbeda pada saat terdapat dua *vertex* dengan *cost* yang sama mengakibatkan perbedaan total *cost* yang ditempuh.

Hasil yang diberikan sistem adalah rute dengan total *cost* lebih besar dimana saat terjadi pilihan antara O1 dan R2 sistem memilih O1. Hal ini disebabkan sistem melakukan pengurutan secara ascending sehingga pencarian rute pada id_user ziya menjadi tidak optimal.

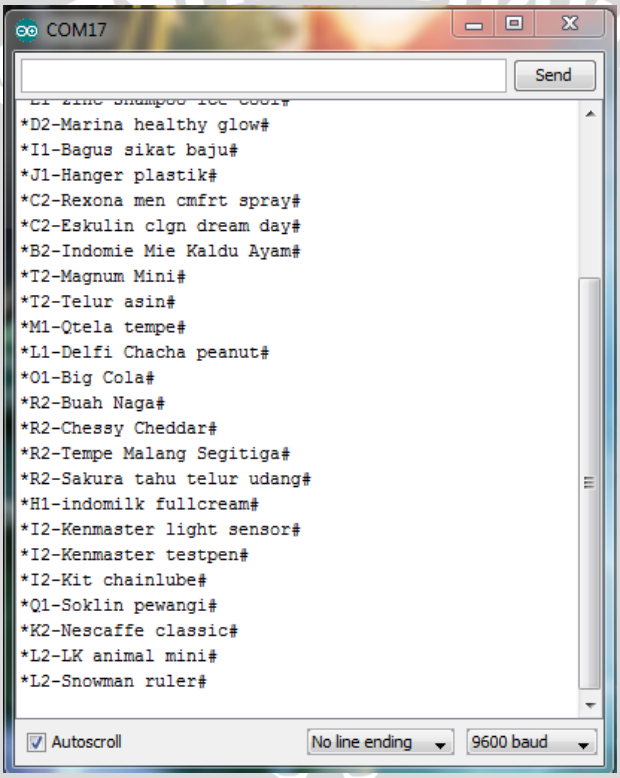
5.4 Pengujian dan Analisis Komunikasi

5.4.1 Pengujian Komunikasi Data dari *Server* ke Arduino

Pengujian komunikasi data dari komputer ke arduino dilakukan untuk mengetahui keberhasilan arduino dalam menerima dan merespon data yang

dikirim dari *server* berdasarkan program yang telah dimuat. Pengujian dilakukan dengan mengirimkan rute belanja 30 pengguna. Dalam pengujian ini arduino akan menampilkan data yang dikirimkan server melalui serial monitor.

Pengujian dilakukan dengan melakukan pengiriman data rute hasil yang sudah diolah pada aplikasi untuk ditampilkan pada modul *wireless*. Data yang dikirim bertipe string dengan jumlah maksimal 50 karakter dan jumlah maksimal 70 data. Pengiriman dilakukan menggunakan fungsi show.message pada Delphi untuk melihat data yang berhasil dikirim. Data yang sudah dikirim ditampilkan pada Serial Monitor Arduino untuk melihat jumlah data yang berhasil diterima.



Gambar 5. 5 Tampilan Pengiriman Data dari Server ke Arduino

Pengiriman dilakukan berdasarkan pilihan belanja id_user. Hasil pengujian selengkapnya sebagai berikut:

- Aji : data terkirim 15, data diterima 15, data hilang 0.
- Azmil : data terkirim 23, data diterima 23, data hilang 0.
- Adip : data terkirim 27, data diterima 27, data hilang 0.
- Bagus : data terkirim 15, data diterima 15, data hilang 0.
- Deby : data terkirim 65, data diterima 48, data hilang 17.

- Dhyk : data terkirim 48, data diterima 46, data hilang 2.
- Dina : data terkirim 32, data diterima 32, data hilang 0.
- Fatih : data terkirim 42, data diterima 42, data hilang 0.
- Fitria : data terkirim 37, data diterima 37, data hilang 0.
- Firdha : data terkirim 35, data diterima 35, data hilang 0.
- Freddy : data terkirim 28, data diterima 28, data hilang 0.
- Hariansah : data terkirim 36, data diterima 36, data hilang 0.
- Hutomo : data terkirim 27, data diterima 27, data hilang 0.
- Ika : data terkirim 67, data diterima 46, data hilang 21.
- Jenar : data terkirim 26, data diterima 26, data hilang 0.
- Mawar : data terkirim 44, data diterima 44, data hilang 0.
- Mes : data terkirim 14, data diterima 14, data hilang 0.
- Mikko : data terkirim 28, data diterima 28, data hilang 0.
- Mimi : data terkirim 44, data diterima 44, data hilang 0.
- Nova : data terkirim 37, data diterima 37, data hilang 0.
- Panji : data terkirim 22, data diterima 22, data hilang 0.
- Rahmat : data terkirim 35, data diterima 35, data hilang 0.
- Rasid : data terkirim 11, data diterima 11, data hilang 0.
- Rifa : data terkirim 29, data diterima 29, data hilang 0.
- Ririn : data terkirim 34, data diterima 34, data hilang 0.
- Rizal : data terkirim 35, data diterima 35, data hilang 0.
- Setyan : data terkirim 28, data diterima 28, data hilang 0.
- Suhartini : data terkirim 43, data diterima 43, data hilang 0.
- Tiyo : data terkirim 31, data diterima 31, data hilang 0.
- Ziya : data terkirim 22, data diterima 22, data hilang 0.

5.4.2 Analisis Komunikasi Data dari *Server* ke Arduino

Dalam pengujian ini, arduino dapat menampilkan hasil rute yang dikirimkan *server*. Dari hasil pengujian dapat diketahui bahwa proses pengirim berhasil dilakukan 27 kali pada 30 percobaan. Arduino sukses menerima data dari *server* dengan persentasi 90%. Jumlah data yang dikirimkan adalah 983 data ,

jumlah data yang diterima adalah 941 data dan jumlah data yang hilang adalah 42 data.

BAB VI
KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan rumusan masalah yang ada, hasil perancangan, implementasi dan pengujian yang dilakukan, maka dapat diambil kesimpulan sebagai berikut:

1) Sistem yang dirancang dapat memberikan rute terpendek untuk memperoleh seluruh barang belanja yang dipilih berdasarkan *rute* yang dibuat dengan akurasi rute dengan *cost* terpendek yang dilewati sebesar 100%.
2) Ketepatan rute yang dihasilkan sistem dalam menentukan jarak terpendek adalah 67%. Hasil didapatkan dari proses pembandingan antara hasil aplikasi dengan perhitungan manual total *cost* yang dilewati menuju seluruh daftar barang belanja yang dipilih.
3) Komunikasi antara modul *wireless* dengan *server* dapat dilakukan menggunakan zigbee dan arduino uno dengan tingkat keberhasilan pengiriman 90% dari 30 percobaan.

6.2 Saran

Saran yang dapat diberikan untuk pengembangan perangkat lunak ini antara lain:

1) Untuk mengolah rute yang lebih banyak diperlukan memori yang lebih besar pada mikrokontroller.
2) Diperlukan analisiss protokol komunikasi yang sesuai dalam pengembangan sistem jika ingin diimplementasikan pada dunia nyata untuk mengatur kepadatan data yang dikirim.
3) Diperlukan skenario pemetaan yang berbeda atau algoritma yang lebih kompleks dalam pencarian rute terpendek dimana pertimbangan dilakukan dengan melihat total *cost* seluruh pilihan barang belanja, tidak hanya satu rute selanjutnya.

DAFTAR PUSTAKA

Ali, Zeeshan. Reena, Sonkusare. (2013). "RFID Based Smart Shopping and Billing". International Journal of Advanced Research and Communication Engineering (IJARCCE). 2(12). 2-4.

Badamasi, Yusuf Abdullahi. (2014). "The Working Principle of an Arduino". Institute of Electrical and Electronic Engineering (IEEE). 7(21). 76-78.

Choi, S.H. Y.X. Yang, B. Yang, H.H. Cheung. (2015). "Item-level RFID for Enhancement of Customer Shopping Experience in Apparel Retail". Elsevier. 5(71). 10-23.

Demirkhan, Haluk, Jim, Spohrer. (2014)." Journal of Retailing and Consumer Services". 27th ed. Washington : Elsevier. pp 860-868.

Djuandi, Feri. (2011). "Pengenalan Arduino". Indonesia: Tobuku. p8.

Fitria, Apri Triansyah. (2013). "Implementasi Algoritma Dijkstra Dalam Aplikasi Untuk Menentukan Lintasan Terpendek Jalan Darat Antar Kota di Sumatera Bagian Selatan". Jurnal Sistem Informasi (JSI). 5(2). 611-621.

Katia. (2015). "Arduino Board Uno". Retrieved from Arduino: (http://www.arduino.cc/en/main/arduinoBoardUno)

Widiasrini, P., & Ferawati, T. R. (2005)." ZIGBEE: Komunikasi Wireless Berdaya Rendah". Jakarta : Universitas Gunadarma. 3ed. Pp 30-35.

LAMPIRAN

Lampiran 1

Lampiran 1 menjelaskan mengenai proses pengujian dan analisa hasil rute. Hasil perbandingan kolom perhitungan manual merupakan hasil perhitungan dari pencarian rute dengan total *cost* terkecil secara manual sementara kolom aplikasi merupakan hasil dari pencarian rute melalui dengan total *cost* terkecil yang dilakukan aplikasi. Lokasi barang belanja merupakan daftar lokasi barang belanja yang dipilih oleh *user* yang akan diolah untuk mendapatkan rute dengan total *cost* terpendek.

Id User	Lokasi Barang Belanja	Hasil Perbandingan			
		Perhitungan Manual		Aplikasi	
		RakTerdekat	Total *Cost*	RakTerdekat	Total*Cost*
Bagus	S1	A1	0	A1	0
	R1	B1	2	B1	2
	T2	D1	6	D1	6
	A1	E1	8	E1	8
	B2	S1	20	S1	20
	C2	R1	22	R1	22
	D1	T2	40	T2	40
	E1	L1	44	L1	44
	I1	B2	48	B2	48
	I2	C2	50	C2	50
	K2	I1	54	I1	54
	L1	H1	56	H1	56
	P1	I2	72	I2	72
	Q1	Q1	74	Q1	74
	B1	K2	78	K2	78
	R2	P1	86	P1	86

Id User	Lokasi Barang Belanja	Hasil Perbandingan			
		Perhitungan Manual		Aplikasi	
		RakTerdekat	Total *Cost*	RakTerdekat	Total *Cost*
Adip	N1	A1	0	A1	0
	A1	B1	2	B1	2
	B1	C1	4	C1	4
	T1	D1	6	D1	6
	R2	E1	8	E1	8
	T2	T1	18	T1	18
	R1	R1	22	R1	22
	K2	R2	36	R2	36
	P1	T2	40	T2	40
	C1	A2	42	A2	42
	A2	B2	44	B2	44
	B2	K1	46	K1	46
	K1	C2	50	C2	50
	D1	N1	60	N1	60
	E1	P1	76	P1	76
	Q1	K2	82	K2	82
	I2	J2	84	J2	84
	C2	I2	86	I2	86
	J2	Q1	88	Q1	88

Id User	Lokasi Barang Belanja	Hasil Perbandingan			
		Perhitungan Manual		Aplikasi	
		RakTerdekat	Total *Cost*	RakTerdekat	Total *Cost*
Rizal	T1	A1	0	A1	0
	A1	T1	2	T1	2
	R2	S1	4	S1	4
	R1	R1	6	R1	6
	H1	B1	14	B1	14
	S1	C1	16	C1	16
	C1	E1	20	E1	20
	B2	H1	36	H1	36
	K1	I1	38	I1	38
	D2	D2	40	D2	40
	E1	C2	42	C2	42
	I1	B2	44	B2	44
	Q1	K1	46	K1	46
	I2	R2	56	R2	56
	B1	J2	82	J2	82
	C2	I2	84	I2	84
	J2	Q1	86	Q1	86

Id User	Lokasi Barang Belanja	Hasil Perbandingan			
		Perhitungan Manual		Aplikasi	
		RakTerdekat	Total *Cost*	RakTerdekat	Total *Cost*
Rahmat	R2	B1	2	B1	2
	T2	C1	4	C1	4
	R1	D1	6	D1	6
	P1	E1	6	E1	8
	H1	S1	18	S1	20
	S1	R1	20	R1	22
	C1	R2	34	R2	36
	A2	T2	36	T2	40
	B2	A2	38	A2	42
	M1	B2	40	B2	44
	M2	C2	42	K1	46
	K1	D2	44	M1	50
	B1	G1	48	C2	58
	D2	H1	50	D2	60
	D1	K1	56	G1	64
	G1	M1	60	H1	66
	E1	M2	74	I2	82
	Q1	P1	78	Q1	84
	I2	Q1	86	J2	86
	C2	J2	88	M2	92
	J2	I2	90	P1	94

Id User	Lokasi Barang Belanja	Hasil Perbandingan			
		Perhitungan Manual		Aplikasi	
		RakTerdekat	Total *Cost*	RakTerdekat	Total *Cost*
Azmil	T1	T1	2	T1	2
	T2	S1	4	S1	4
	O1	R1	6	R1	6
	P1	B1	14	B1	14
	R2	C1	16	C1	16
	R1	D1	18	D1	18
	B2	E1	20	E1	20
	S1	G1	36	D2	36
	G1	D2	40	C2	38
	D1	C2	42	B2	40
	D2	B2	44	T2	44
	E1	T2	48	R2	52
	C1	R2	52	O1	54
	Q1	O1	54	G1	70

	I2	P1	70	Q1	88
	B1	Q1	78	I2	90
	C2	I2	80	P1	100
Id User	Lokasi Barang Belanja	Hasil Perbandingan			
		Perhitungan Manual		Aplikasi	
		RakTerdekat	Total *Cost*	RakTerdekat	Total *Cost*
Rifa	N1	A1	0	A1	0
	A1	B1	2	B1	2
	T1	C1	4	C1	4
	R2	D1	6	D1	6
	T2	E1	8	E1	8
	R1	T1	18	T1	18
	P1	R1	22	R1	22
	H1	R2	36	R2	36
	M2	T2	40	T2	40
	D1	N1	44	N1	44
	D2	C2	54	C2	54
	E1	D2	56	D2	56
	C1	F1	59	F1	59
	Q1	H1	62	H1	62
	I2	I2	78	I2	78
	B1	Q1	80	Q1	80
	F1	P1	88	M2	88
	C2	M2	90	P1	90
Id User	Lokasi Barang Belanja	Hasil Perbandingan			
		Perhitungan Manual		Aplikasi	
		RakTerdekat	Total *Cost*	RakTerdekat	Total *Cost*
Suhartini	T1	A1	0	A1	0
	A1	T1	2	T1	2
	A2	S1	4	S1	4
	R2	R1	6	R1	6
	T2	B1	14	B1	14
	R1	C1	16	C1	16
	B1	E1	20	E1	20
	C1	F1	35	F1	35
	B2	I1	40	I1	40
	L1	J1	42	J1	42
	M2	K1	44	K1	44
	S1	B2	46	B2	46
	K1	A2	48	A2	48
	E1	L1	50	L1	50
	J1	T2	54	T2	54
	I1	R2	58	R2	58
	Q1	M2	78	M2	78
	I2	L2	80	L2	80
	F1	J2	84	J2	84
	J2	I2	86	I2	86
	L2	Q1	88	Q1	88
Id User	Lokasi Barang Belanja	Hasil Perbandingan			
		Perhitungan Manual		Aplikasi	
		RakTerdekat	Total *Cost*	Rak Terdekat	Total *Cost*
Aji	A1	A1	0	A1	0
	B1	T1	2	T1	2
	D1	S1	4	S1	4
	D2	R1	6	R1	6
	E1	B1	14	B1	14
	K2	D1	18	D1	18
	L1	E1	20	E1	20
	R1	D2	36	D2	36
	R2	L1	44	L1	44
	S1	T2	48	T2	48
	T1	R2	52	R2	52

	T2	K2	76	K2	76
Id User	Lokasi Barang Belanja	Hasil Perbandingan			
		Perhitungan Manual		Aplikasi	
		RakTerdekat	Total *Cost*	Rak Terdekat	Total *Cost*
Deby	A1	A1	0	A1	0
	A2	B1	2	B1	2
	B1	C1	4	C1	4
	B2	D1	6	D1	6
	C1	E1	8	E1	8
	C2	T1	18	T1	18
	D1	S1	20	S1	20
	E1	R1	22	R1	22
	F1	R2	36	R2	36
	G1	O1	38	O1	38
	H1	T2	42	T2	42
	I1	A2	44	A2	44
	I2	L1	46	L1	46
	J2	K1	48	K1	48
	K1	B2	50	B2	50
	K2	C2	52	C2	52
	L1	I1	56	I1	56
	L2	H1	58	H1	58
	M2	G1	60	G1	60
	O1	F1	61	F1	61
	P1	I2	80	I2	80
	Q1	J2	82	J2	82
	R1	K2	84	K2	84
	R2	L2	86	L2	86
	S1	M2	88	M2	88
Id User	Lokasi Barang Belanja	Hasil Perbandingan			
		Perhitungan Manual		Aplikasi	
		RakTerdekat	Total *Cost*	Rak Terdekat	Total *Cost*
Dhyk	A1	A1	0	A1	0
	B1	B1	2	B1	2
	C1	C1	4	C1	4
	C2	D1	6	D1	6
	D1	E1	8	E1	8
	D2	S1	20	S1	20
	E1	R1	22	R1	22
	G1	R2	36	R2	36
	H2	N1	40	N1	40
	I1	M1	42	M1	42
	I2	T2	44	T2	44
	J2	L1	48	L1	48
	K1	K1	50	K1	50
	K2	C2	54	C2	54
	L1	D2	56	D2	56
	L2	I1	58	I1	58
	M1	G1	62	G1	62
	M2	H2	78	H2	78
	N1	I2	80	I2	80
	P1	J2	82	J2	82
	Q1	K2	84	K2	84
	R1	L2	86	L2	86
	R2	M2	88	M2	88
	S1	P1	90	P1	90
	T2	Q1	98	Q1	98
Id User	Lokasi Barang Belanja	Hasil Perbandingan			
		Perhitungan Manual		Aplikasi	
		RakTerdekat	Total *Cost*	Rak Terdekat	Total *Cost*
Dina	A2	B1	2	B1	2
	B1	C1	4	C1	4

	B2	D1	6	D1	6
	C1	E1	8	E1	8
	C2	S1	20	S1	20
	D1	R1	22	R1	22
	D2	R2	36	R2	36
	E1	T2	40	T2	40
	F1	A2	42	A2	42
	I1	L1	44	L1	44
	I2	K1	46	K1	46
	K1	B2	48	B2	48
	K2	C2	50	C2	50
	L1	D2	52	D2	52
	L2	I1	54	I1	54
	P1	F1	59	F1	59
	Q1	I2	78	I2	78
	R1	Q1	80	Q1	80
	R2	K2	84	K2	84
	S1	L2	86	L2	86

Id User	Lokasi Barang Belanja	Hasil Perbandingan			
		Perhitungan Manual		Aplikasi	
		RakTerdekat	Total *Cost*	Rak Terdekat	Total *Cost*
Fatih	T1	A1	0	A1	0
	A1	T1	2	T1	2
	A2	S1	4	S1	4
	B1	R1	6	R1	6
	R2	B1	14	B1	14
	T2	C1	16	C1	16
	R1	D1	18	D1	18
	O1	E1	20	E1	20
	P1	I1	38	I1	38
	K2	J1	40	J1	40
	B2	B2	44	B2	44
	C1	A2	46	A2	46
	L1	L1	48	L1	48
	M1	M1	50	M1	50
	S1	T2	52	T2	52
	D1	R2	56	O1	56
	E1	O1	58	R2	58
	J1	P1	76	P1	78
	I1	L2	80	L2	82
	Q1	K2	82	K2	84
	I2	J2	84	J2	86
	L2	I2	86	I2	88
	J2	Q1	88	Q1	90

Id User	Lokasi Barang Belanja	Hasil Perbandingan			
		Perhitungan Manual		Aplikasi	
		RakTerdekat	Total *Cost*	Rak Terdekat	Total *Cost*
Firdha	A1	A1	0	A1	0
	B1	B1	2	B1	2
	C1	C1	4	C1	4
	C2	D1	6	D1	6
	D1	E1	8	E1	8
	D2	T1	18	T1	18
	E1	S1	20	S1	20
	F1	R1	22	R1	22
	G1	R2	36	R2	36
	H1	O1	38	O1	38
	I1	L1	44	L1	44
	I2	C2	60	C2	60
	J2	D2	62	D2	62
	L1	I1	64	I1	64
	L2	H1	66	H1	66

	O1	G1	68	G1	68
	P1	F1	69	F1	69
	Q1	I2	88	I2	88
	R1	J2	90	J2	90
	R2	Q1	92	Q1	92
	S1	L2	98	L2	98
	T1	P1	102	P1	102

Id User	Lokasi Barang Belanja	Hasil Perbandingan			
		Perhitungan Manual		Aplikasi	
		RakTerdekat	Total *Cost*	Rak Terdekat	Total *Cost*
Fitria	A2	T1	2	T1	2
	C1	S1	4	S1	4
	C2	R1	6	R1	6
	D1	C1	16	C1	16
	D2	D1	18	D1	18
	E1	E1	20	E1	20
	F1	F1	35	F1	35
	G1	G1	36	G1	36
	H1	H1	38	H1	38
	I1	I1	40	I1	40
	I2	D2	42	D2	42
	L1	C2	44	C2	44
	L2	A2	46	A2	46
	M2	R2	48	R2	48
	O1	M2	68	M2	68
	Q1	L2	70	L2	70
	R1	I2	76	I2	76
	R2	Q1	78	Q1	78

Id User	Lokasi Barang Belanja	Hasil Perbandingan			
		Perhitungan Manual		Aplikasi	
		Rak Terdekat	Total *Cost*	Rak Terdekat	Total *Cost*
Freddy	A1	A1	0	A1	0
	A2	T1	2	T1	2
	B1	S1	4	S1	4
	C1	R1	6	R1	6
	C2	B1	14	B1	14
	D1	C1	16	C1	16
	E1	D1	18	D1	18
	I2	E1	20	E1	20
	J1	C2	38	C2	38
	L2	J1	40	J1	40
	M1	A2	46	A2	46
	Q1	T2	48	T2	48
	R1	M1	50	M1	50
	R2	R2	56	R2	56
	S1	L2	78	L2	78
	T1	I2	84	I2	84
	T2	Q1	86	Q1	86

Id User	Lokasi Barang Belanja	Hasil Perbandingan			
		Perhitungan Manual		Aplikasi	
		Rak Terdekat	Total *Cost*	Rak Terdekat	Total *Cost*
Hariansah	A1	A1	0	A1	0
	A2	B1	2	B1	2
	B1	C1	4	C1	4
	B2	E1	8	E1	8
	C1	S1	20	S1	20
	C2	R1	22	R1	22
	D2	R2	36	R2	36
	E1	O1	38	O1	38
	H1	N1	40	N1	40
	I1	L1	44	L1	44
	I2	A2	46	A2	46

	K1	B2	48	B2	48
	K2	C2	50	C2	50
	L1	D2	52	D2	52
	L2	I1	54	I1	54
	N1	H1	56	H1	56
	O1	K1	62	K1	62
	P1	T2	68	T2	68
	R1	L2	86	L2	86
	R2	K2	88	K2	88
	S1	I2	92	I2	92
	T2	P1	102	P1	102

Id User	Lokasi Barang Belanja	Hasil Perbandingan			
		Perhitungan Manual		Aplikasi	
		Rak Terdekat	Total *Cost*	RakTerdekat	Total *Cost*
Hutomo	A2	B1	2	B1	2
	B1	C1	4	C1	4
	C1	D1	6	D1	6
	C2	E1	8	E1	8
	D1	S1	20	S1	20
	E1	R1	22	R1	22
	I2	R2	36	R2	36
	K1	O1	38	O1	38
	L2	T2	42	T2	42
	O1	A2	44	A2	44
	Q1	C2	48	C2	48
	R1	K1	52	K1	52
	R2	L2	68	L2	68
	S1	I2	74	I2	74
	T2	Q1	76	Q1	76

Id User	Lokasi Barang Belanja	Hasil Perbandingan			
		Perhitungan Manual		Aplikasi	
		RakTerdekat	Total *Cost*	RakTerdekat	Total *Cost*
Ika	A1	A1	0	A1	0
	A2	B1	2	B1	2
	B1	C1	4	C1	4
	C1	D1	6	D1	6
	C2	E1	8	E1	8
	D1	T1	18	T1	18
	E1	S1	20	S1	20
	G1	R1	22	R1	22
	I1	R2	36	R2	36
	I2	O1	38	O1	38
	J2	T2	42	T2	42
	K1	A2	44	A2	44
	K2	L1	46	L1	46
	L1	K1	48	K1	48
	L2	C2	52	C2	52
	M2	I1	56	I1	56
	O1	G1	60	G1	60
	P1	I2	78	I2	78
	Q1	Q1	80	J2	80
	R1	J2	82	K2	82
	R2	K2	84	L2	84
	S1	L2	86	M2	86
	T1	M2	88	P1	88
	T2	P1	90	Q1	96

Id User	Lokasi Barang Belanja	Hasil Perbandingan			
		Perhitungan Manual		Aplikasi	
		RakTerdekat	Total *Cost*	Rak Terdekat	Total*Cost*
Jenar	A2	B1	4	B1	4
	B1	C1	6	C1	6

	B2	D1	8	D1	8
	C1	E1	10	E1	10
	D1	T1	20	T1	20
	E1	S1	22	S1	22
	I1	R1	24	R1	24
	I2	R2	38	R2	38
	J1	N1	42	N1	42
	J2	T2	46	T2	46
	L1	A2	48	A2	48
	L2	B2	50	B2	50
	N1	J1	54	J1	54
	P1	I1	56	I1	56
	R1	L1	62	L1	62
	R2	L2	76	L2	76
	S1	P1	80	P1	80
	T1	J2	88	J2	88
	T2	I2	90	I2	90
Id User	Lokasi Barang Belanja	Hasil Perbandingan			
		Perhitungan Manual		Aplikasi	
		RakTerdekat	Total *Cost*	Rak Terdekat	Total *Cost*
Mawar	A1	A1	0	A1	0
	A2	T1	2	T1	2
	B1	R1	6	R1	6
	B2	B1	14	B1	14
	C1	C1	16	C1	16
	C2	D1	18	D1	18
	D1	E1	20	E1	20
	D2	F1	35	F1	35
	E1	G1	36	G1	36
	F1	I1	40	I1	40
	G1	D2	42	D2	42
	I1	C2	44	C2	44
	I2	B2	46	B2	46
	J2	K1	48	K1	48
	K1	L1	50	L1	50
	K2	A2	52	A2	52
	L1	T2	54	T2	54
	L2	R2	58	R2	58
	O1	O1	60	O1	60
	P1	P1	78	P1	78
	Q1	L2	82	L2	82
	R1	K2	84	K2	84
	R2	J2	86	J2	86
	T1	I2	88	I2	88
	T2	Q1	90	Q1	90
Id User	Lokasi Barang Belanja	Hasil Perbandingan			
		Perhitungan Manual		Aplikasi	
		RakTerdekat	Total *Cost*	Rak Terdekat	Total*Cost*
Mes	A2	B1	2	B1	2
	B1	C1	4	C1	4
	B2	E1	8	E1	8
	C1	R1	22	R1	22
	C2	R2	36	R2	36
	D2	O1	38	O1	38
	E1	A2	44	A2	44
	G1	B2	46	B2	46
	K2	C2	48	C2	48
	O1	D2	50	D2	50
	Q1	G1	54	G1	54
	R1	Q1	72	Q1	72
	R2	K2	76	K2	76

Id User	Lokasi Barang Belanja	Hasil Perbandingan			
		Perhitungan Manual		Aplikasi	
		RakTerdekat	Total *Cost*	Rak Terdekat	Total *Cost*
Mikko	B1	B1	2	B1	2
	B2	C1	4	C1	4
	C1	E1	8	E1	8
	C2	S1	20	S1	20
	D2	R1	22	R1	22
	I1	R2	36	R2	36
	I2	O1	38	O1	38
	J2	L1	44	L1	44
	K1	K1	46	K1	46
	L1	B2	48	B2	48
	L2	C2	50	C2	50
	O1	D2	52	D2	52
	Q1	I1	54	I1	54
	R1	H1	56	H1	56
	R2	I2	72	I2	72
	S1	Q1	74	J2	74
	E1	J2	76	Q1	76
	H1	L2	80	L2	82

Id User	Lokasi Barang Belanja	Hasil Perbandingan			
		Perhitungan Manual		Aplikasi	
		RakTerdekat	Total *Cost*	Rak Terdekat	Total *Cost*
Mimi	A1	A1	0	A1	0
	A2	B1	2	T1	2
	B1	C1	6	B1	6
	B2	D1	8	C1	8
	C1	E1	10	D1	10
	C2	T1	20	E1	12
	D1	R1	24	R1	26
	D2	R2	38	R2	40
	E1	T2	42	T2	44
	F1	A2	44	A2	46
	G1	B2	46	B2	48
	H1	C2	48	C2	50
	I1	D2	50	D2	52
	J1	I1	52	I1	54
	J2	H1	54	H1	56
	K2	G1	56	G1	58
	L2	F1	57	F1	59
	P1	J1	64	J1	66
	Q1	J2	78	J2	80
	R1	Q1	80	K2	82
	R2	K2	84	L2	84
	T1	L2	86	P1	88
	T2	P1	90	Q1	96

Id User	Lokasi Barang Belanja	Hasil Perbandingan			
		Perhitungan Manual		Aplikasi	
		RakTerdekat	Total *Cost*	Rak Terdekat	Total *Cost*
Nova	A1	A1	0	A1	0
	B1	B1	2	B1	2
	B2	C1	4	C1	4
	C1	D1	6	D1	6
	C2	E1	8	E1	8
	D1	T1	18	T1	18
	E1	S1	20	S1	20
	F1	R1	22	R1	22
	G1	R2	36	R2	36
	I2	T2	40	T2	40
	J2	B2	44	B2	44
	K1	C2	42	C2	42

	L1	K1	46	K1	46
	L2	L1	48	L1	48
	P1	G1	58	G1	58
	Q1	F1	59	F1	59
	R1	Q1	78	Q1	78
	R2	I2	80	I2	80
	S1	J2	82	J2	82
	T1	L2	86	L2	86
	T2	P1	90	P1	90

Id User	Lokasi Barang Belanja	Hasil Perbandingan			
		Perhitungan Manual		Aplikasi	
		Rak Terdekat	Total*Cost*	Rak Terdekat	Total *Cost*
Panji	B1	B1	2	B1	2
	B2	D1	6	D1	6
	C2	E1	8	E1	8
	D1	T1	18	T1	18
	E1	S1	20	S1	20
	H1	R1	22	R1	22
	I1	R2	36	R2	36
	I2	B2	44	B2	44
	K2	C2	46	C2	46
	L2	I1	50	I1	50
	P1	H1	52	H1	52
	Q1	I2	68	I2	68
	R1	Q1	70	Q1	70
	R2	K2	74	K2	74
	S1	L2	76	L2	76
	T1	P1	80	P1	80

Id User	Lokasi Barang Belanja	Hasil Perbandingan			
		Perhitungan Manual		Aplikasi	
		RakTerdekat	Total *Cost*	RakTerdekat	Total*Cost*
Rasid	A1	A1	0	A1	0
	D2	T1	2	T1	2
	E1	R1	6	R1	6
	H1	E1	20	E1	20
	L2	D2	36	D2	36
	P1	H1	40	H1	40
	Q1	O1	54	O1	54
	R1	R2	56	R2	56
	R2	P1	76	P1	76
	T1	L2	80	L2	80
	O1	Q1	86	Q1	86

Id User	Lokasi Barang Belanja	Hasil Perbandingan			
		Perhitungan Manual		Aplikasi	
		RakTerdekat	Total *Cost*	RakTerdekat	Total*Cost*
Ririn	A1	A1	0	A1	0
	A2	T1	2	T1	2
	B1	S1	4	S1	4
	B2	R1	6	R1	6
	C1	B1	14	B1	14
	C2	C1	16	C1	16
	D1	D1	18	D1	18
	E1	E1	20	E1	20
	F1	F1	35	F1	35
	I2	C2	40	C2	40
	J2	B2	42	B2	42
	K1	A2	44	A2	44
	L2	T2	46	T2	46
	Q1	R2	50	R2	50
	R1	K1	60	K1	60
	R2	L2	76	J2	76
	S1	J2	80	I2	78

	T1	I2	82	Q1	80
	T2	Q1	84	L2	86

Id User	Lokasi Barang Belanja	Hasil Perbandingan			
		Perhitungan Manual		Aplikasi	
		Rak Terdekat	Total *Cost*	Rak Terdekat	Total *Cost*
Setyan	A1	A1	0	A1	0
	A2	T1	2	T1	2
	B1	S1	4	S1	4
	B2	R1	6	R1	6
	C1	B1	14	B1	14
	C2	C1	16	C1	16
	D1	D1	18	D1	18
	D2	E1	20	E1	20
	E1	D2	36	D2	36
	G1	C2	38	C2	38
	I1	B2	40	B2	40
	I2	A2	42	A2	42
	K1	K1	46	K1	46
	L2	I1	50	I1	50
	Q1	G1	54	G1	54
	R1	R2	70	R2	70
	R2	L2	92	L2	92
	S1	I2	98	I2	98
	T1	Q1	100	Q1	100

Id User	Lokasi Barang Belanja	Hasil Perbandingan			
		Perhitungan Manual		Aplikasi	
		Rak Terdekat	Total *Cost*	Rak Terdekat	Total *Cost*
Tiyo	B1	T1	2	T1	2
	B2	S1	4	S1	4
	C1	R1	6	R1	6
	C2	B1	14	B1	14
	D2	C1	16	C1	16
	E1	E1	20	E1	20
	H1	D2	36	D2	36
	I1	I1	38	I1	38
	I2	H1	40	J1	40
	J1	J1	44	C2	42
	K2	C2	46	B2	44
	L1	B2	48	T2	48
	L2	T2	52	M1	50
	M1	M1	54	L1	52
	O1	L1	56	O1	58
	Q1	O1	62	R2	60
	R1	R2	64	H1	76
	R2	L2	86	I2	92
	S1	K2	88	Q1	96
	T1	I2	92	K2	100
	T2	Q1	94	L2	102

Id User	Lokasi Barang Belanja	Hasil Perbandingan			
		Perhitungan Manual		Aplikasi	
		Rak Terdekat	Total *Cost*	Rak Terdekat	Total *Cost*
Ziya	C2	S1	4	S1	4
	D1	R1	6	R1	6
	D2	D1	18	D1	18
	E1	E1	20	E1	20
	F1	F1	35	F1	35
	I2	D2	38	D2	38
	K2	C2	40	C2	40
	L1	L1	46	L1	46
	L2	T2	50	T2	50
	O1	R2	54	O1	54

	P1	O1	56	R2	56
	R1	P1	74	P1	76
	R2	L2	78	L2	80
	S1	K2	80	K2	82
	T2	I2	84	I2	86

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA