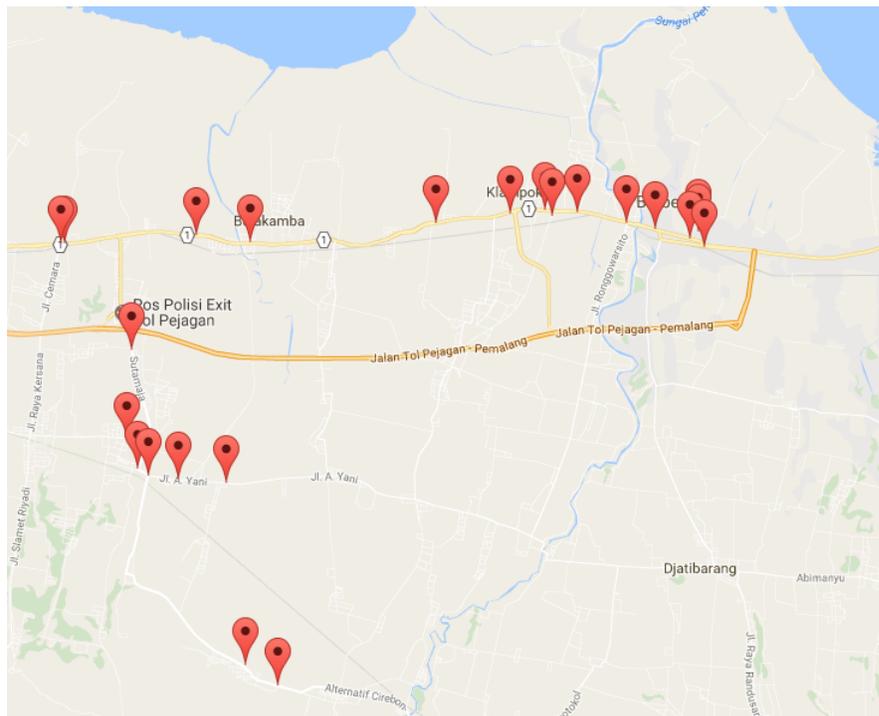


BAB IV

IMPLEMENTASI DAN PEMBAHASAN

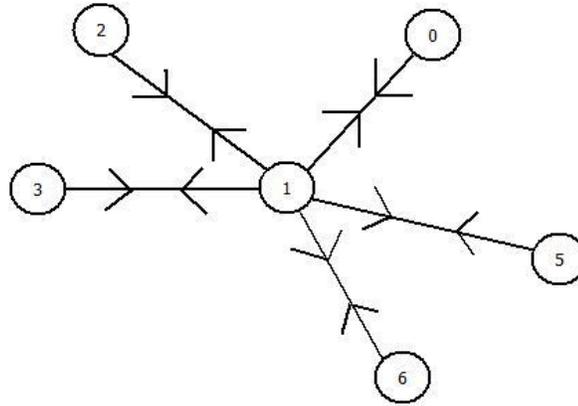
4.1. Implementasi Sistem

Untuk implementasi algoritma Dijkstra, Diperlukan peta yang akan diambil koordinatnya kemudian akan diproses. Berikut adalah peta objek penjualan oleh-oleh khas Kabupaten Brebes :



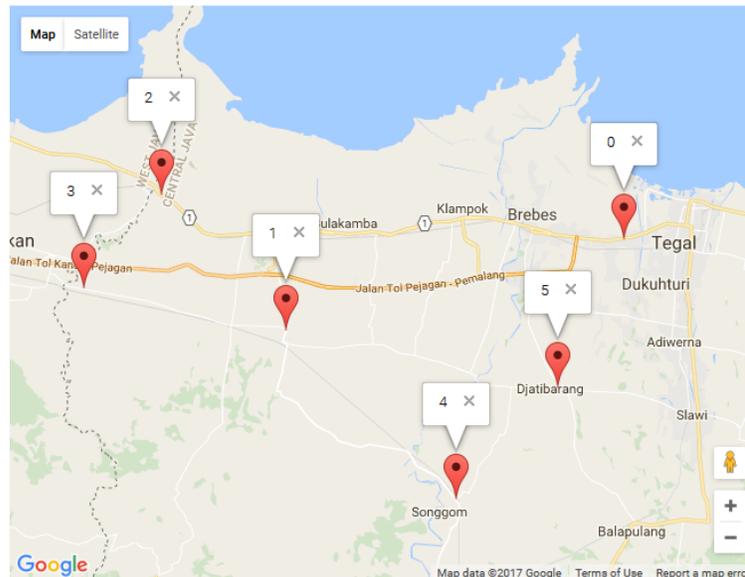
Gambar 4.1 Peta Lokasi Objek Penjualan

Berikut adalah graph node-node yang telah di tentukan dari peta yang dibuat di Kabupaten Brebes dan node ini juga merupakan simpul terluar yang akan di baca oleh algoritma Dijkstra.



Gambar 4.2 Graph Node-Node Pada Peta

Berikut konversi Graph ke dalam Peta

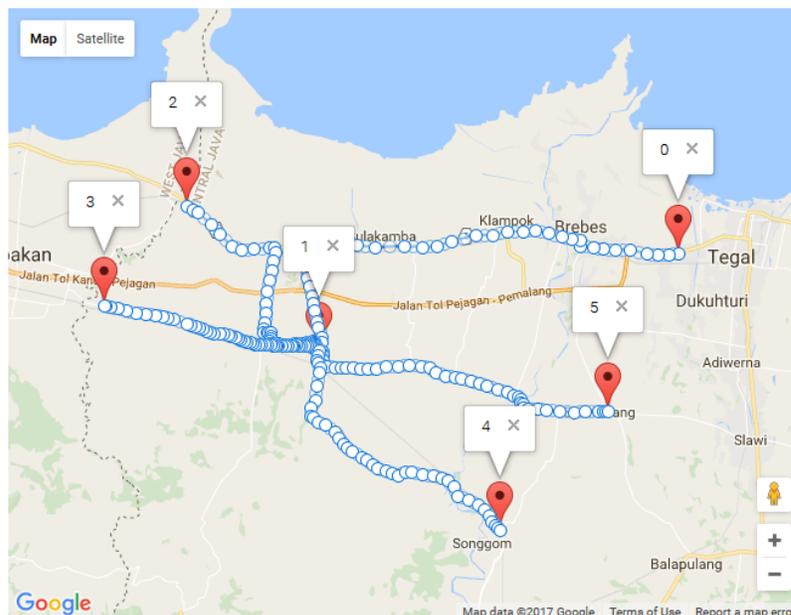


Gambar 4.3 Konversi Graph ke Dalam Peta

Pada Graph diatas node-node nya telah di tentukan sebagai berikut :

Node 0 berada di Kecamatan Brebes,node 2 di Kecamatan Losari ,node 1 di Kecamatan Ketanggungan,node 3 di Kecamatan Kersana,node 4 di Kecamatan Bumiayu dan node 5 di Kecamatan Sitanggal sehingga apabila user meletakkan koordinat diluar graph node-node yang ada maka algoritma Dijkstra hanya membaca di node-node yang sudah ditentukan saja, contoh seperti user meletakkan koordinat di jogja maka yang terbaca oleh algoritma Dijkstra terdekat di node 5 yaitu dikecamatan Bumiayu.

Berikut adalah jalur atau line penghubung antar node-node.



Gambar 4.4 Jalur Penghubung Antar node-node

Pada gambar diatas adalah jalur penghubung setiap node-node yang ada yakni node 0 terhubung ke node 1 membuat jalur dari node 0 ke node 1 demikian juga untuk node-node yang lain dibuat jalur masing-masing seperti gambar di atas sehingga terbentuk rute Angkatan Umum seperti berikut ini :

```

T01=, 0-1, 1-0, 0-2, 2-0, 0-3, 3-0, 0-4, 4-0, 0-5, 5-0,
T02=, 1-0, 0-1, 1-2, 2-1, 1-3, 3-1, 1-4, 4-1, 1-5, 5-1,
T03=, 2-0, 0-2, 2-1, 1-2, 2-3, 3-2, 3-4, 4-3, 3-5, 5-3,
T04=, 3-0, 0-3, 3-1, 1-3, 3-2, 2-3, 3-4, 4-3, 3-5, 5-3,
T05=, 4-0, 0-4, 4-1, 1-4, 4-2, 2-4, 4-3, 3-4, 4-5, 5-4,
T06=, 5-0, 0-5, 5-1, 1-5, 5-2, 2-5, 5-3, 3-5, 5-4, 4-5,

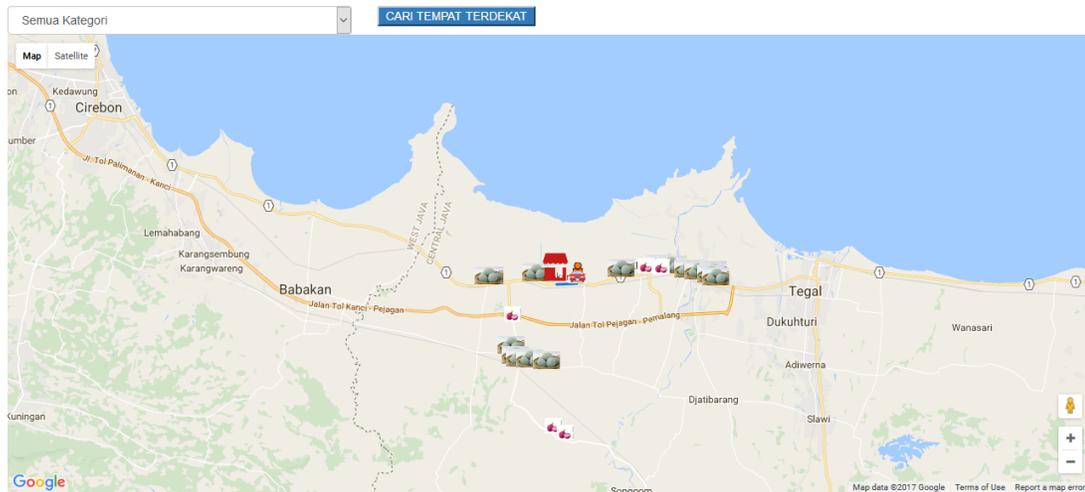
```

Gambar 4.5 Rute Angkatan Umum

Pada gambar diatas di jelaskan bahwa rute angkutan membuat 6 rute jalur Angkatan Umum yaitu T01,T02,T03,T04,T05,T06.contoh penjelasan T01 adalah 0-1 artinya rute dari node 0 ke node 1 dan 1-0 sama sebaliknya dari node 1 ke node 0,1-2 artinya rute dari node 1 ke node 2 dan 2-1 sama sebaliknya dari node 2 ke node 1 demikian untuk seterusnya sama juga yang lain sehingga rute jalur telah ditentukan apabila user meletakkan koordinat di luar rute maka yang di baca algoritma Dijkstra hanya jalur yang paling didekat kordinaat nya dengan jalur rute yang ada.

Berikut implementasi dari contoh diatas dalam peta asli dari lokasi objek penjualan :

SISTEM INFORMASI GEOGRAFI OLEH-OLEH KHAS DI KAB. BREBES

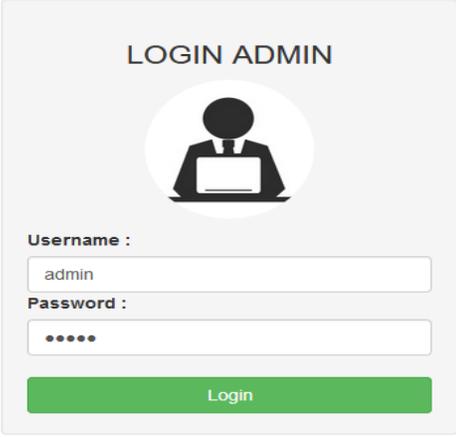


Gambar 4.6 Implementasi Pencarian Rute Terdekat

Pada gambar diatas diketahui bahwa user meletakkan di titik awal koordinat di jalur daerah Kecamatan Bulakamba sedang di jalur tersebut adalah beberapa Toko Telur asin seperti Toko telur asin Raja dan Toko Telur asin Abah jaya dapur dan jalur paling dekat dari user adalah Toko Telur asin Raja.

4.2.1. Halaman Login

Halaman login adalah halaman yang berfungsi sebagai pintu masuk seorang admin ke dalam aplikasi yang nantinya admin akan melakukan aktivitas didalam aplikasi web.



LOGIN ADMIN

Username :
admin

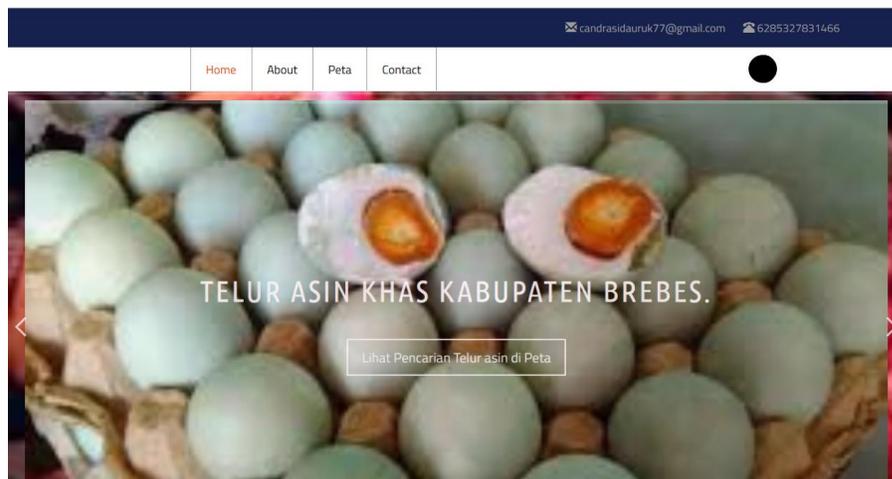
Password :
.....

Login

Gambar 4.7 Halaman Login

4.2.2. Halaman Home

Halaman home adalah halaman yang pertama dilihat oleh pengunjung website ketika membuka website.



Gambar 4.8 Halaman Home

3.2.3. Class Dijkstra

```

class Dijkstra
{
    function jalurTerpendek($arg_graph, $simpulAwal, $simpulTujuan){

        if($simpulAwal == $simpulTujuan){
            return json_encode(['status'=>'error','error'=>'lokasi_anda_sudah_dekat',
                'teks'=>'Lokasi Anda Sudah Dekat','content'=>'']);
        }

        if(!array_key_exists($simpulAwal, $arg_graph) || !array_key_exists($simpulTujuan, $arg_graph)){
            return print_r(json_encode(['status'=>'error','error'=>'simpul_input_tidak_ditemukan',
                'teks'=>"could not find the input : $simpulAwal or $simpulTujuan", 'content'=>'']));
        }
    }
}

```

Gambar 4.8 Class Dijkstra

Script diatas terdapat class Dijkstra yang didalam nya mendeklarasikan function jalurTerpendek dengan 3 parameter yaitu \$arg_graph,\$simpulAwal dan \$simpulTujuan dimana kondisi jika \$simpulAwal == \$simpulTujuan maka “lokasi_anda_sudah_dekat dan jika \$simpulAwal dan \$arg_graph tidak ada maka “simpul_input_tidak_ditemukan”.

3.2.4. Class Get_koordinat_awal_akhir

```

<?php
class Get_koordinat_awal_akhir extends DistanceTo{

    if($indexCoordinate == 0 || $indexCoordinate == $countCoordinate){

        if($indexCoordinate == 0)
        {
            $fix_simpul_awal = $field_simpul_awal;
        }else if($indexCoordinate == $countCoordinate)
        {
            $fix_simpul_akhir = $field_simpul_tujuan;
        }

        $jadi_json['status'] = "tidak_tambah_simpul";
    }

    else
    {
        $select = "SELECT count(id) as jum_id FROM graph where simpul_awal =
        ".$field_simpul_tujuan." and simpul_tujuan = ".$field_simpul_awal;
        $query = mysqli_query($this->koneksi, $select);
        $dobel = mysqli_fetch_array($query, MYSQLI_ASSOC);

        if($dobel['jum_id'] == 1){
            $jadi_json['status'] = "tambah_simpul_double";
        }

        else if($dobel['jum_id'] == 0){
            $jadi_json['status'] = "tambah_simpul_single";
        }
    }
}
?>

```

Gambar 4.9 Get koordinat Awal dan Akhir

Skript pada Gambar 4.9 merupakan pengembalian koordinat awal ke kordinat tujuan yang dibaca oleh algoritma Dijkstra dan setelah didapat baru dihitung jarak antar awal dan tujuan sehingga dapat diketahui mana yang paling dekat jaraknya.

3.2.5. Fungsi CariTerdekat Untuk Menghitung Jarak Terdekat

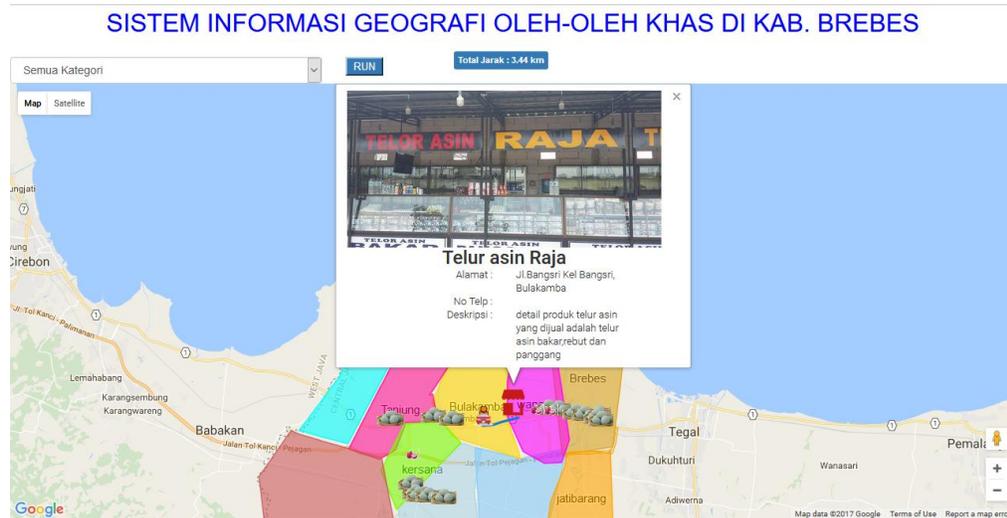
```
function cariTerdekat(lat1,lon1,lat2,lon2) {  
  
    var latit = (lat2-lat1)*(lat2-lat1);  
    var longi = (lon2-lon1)*(lon2-lon1);  
    var d = Math.sqrt(latit+longi);  
  
    return d;  
}
```

Gambar 4.10 Fungsi CariTerdekat Menghitung Jarak Terdekat

Script diatas ini adalah merupakan fungsi DistanceTo yang merupakan bagian dari algoritma dijkstra untuk perhitungan jarak terdekat dari lokasi awal ke lokasi tujuan.

4.2.6. Halaman Lokasi Objek Penjualan

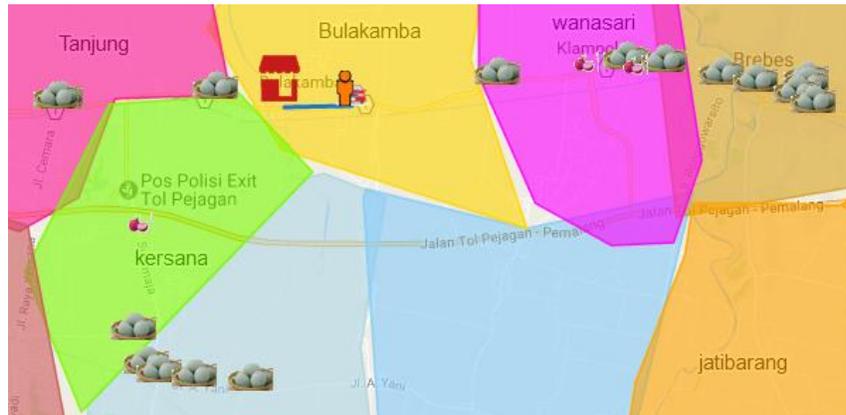
Halaman lokasi objek penjualan adalah halaman yang menampilkan peta kabupaten Brebes dengan lokasi penjualan yang masing-masing ditandai oleh marker dan info window.ketika marker objek Penjualan tersebut diklik maka akan menampilkan Nama Toko,Alamat,Foto,Deskripsi.No Telpon dari lokasi penjualan.



Gambar 4.11 Halaman Lokasi Objek Penjualan

3.2.7 Halaman Rute Jalur Objek Penjualan

Halaman rute jalur objek penjualan menampilkan rute jalur yang paling dekat antar lokasi user ke lokasi objek penjualan yang ada dan yang paling dekat. Pada menu ini user dapat dengan bebas menentukan lokasi awal dan user jika dapat memilih select pilih kategori seperti select pilih semua kategori maka akan muncul semua kategori telur asin dan bawang merah di peta dan setelah itu user mengklik tombol Run maka secara otomatis Algoritma Dijkstra menampilkan rute jalan terdekat ke lokasi tujuan tersebut dengan menggunakan perhitungan algoritma dijkstra.



Gambar 4.12 Halaman Rute Jalur Objek Penjualan

4.3. Pembahasan Sistem

4.3.1 Uji Coba Sistem

Aplikasi Pencarian Oleh-oleh khas Kab.Brebes ini telah diuji dengan menggunakan beberapa kasus. Rekapitulasi pengujian terhadap aplikasi ini dapat dilihat pada Tabel 4.1.

Sebagai contoh kasus pencarian jalur terpendek menggunakan Algoritma Dijkstra,yakni :

Lokasi asal : Bulakamba

Lokasi Tujuan Terdekat : Toko telur asin Abah Jaya Dapur

Jarak terpendek : 1.82 km

Berikut Tabel Perbandingan jarak Program dan Manual :

Tabel 4.1 Tabel Perbandingan jarak Program dan Manual

No	Titik awal dan titik akhir	Hasil Dari Manual	Hasil dari Aplikasi
1	Titik awal : -6.875113,108.951503 Titik akhir : -6.867987, 108.980117	3.28 km	3.44 km
2	Titik awal : -6.873936,108.940465 Titik akhir : -6.874364, 108.922555	1.99 km	1.82 km
3	Titik awal : -6.93245, 108.89255 Titik akhir : -6.93047, 108.8871	0.71 km	0.65 km

Dari hasil perhitungan manual dan aplikasi di dapat hasil dari tiap-tiap sample yang mana sampel 1 selisihnya perbedaannya hanya 0,6 , sampel 2 selisihnya 0.17 dan sampel 3 selisihnya yaitu 0.06. Dan total selisih ketiga sampel yaitu 0.83 sehingga dapat disimpulkan selisih dari perhitungan manual dan aplikasi hanya sedikit saja selisihnya.