

SISTEM PAKAR DIAGNOSA KERUSAKAN MOBIL TOYOTA RUSH MENGUNAKAN METODE BACKWARD CHAINING DAN PENCARIAN RUTE TERDEKAT LOKASI BENGKEL DENGAN ALGORITMA FLOYD WARSHALL

Eric Sugiarto¹, Wijanarto²

Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro

Jl. Nakula No. 5-11 Semarang, Kode Pos 50131, Telp. (024)3515261, 3520165 Fax:3569684

E-mail :samuel.eric.sugiarto@gmail.com¹, wijanarto.udinus@gmail.com²

Abstrak

Kerusakan yang biasa terjadi secara tiba-tiba dan tidak dapat langsung terdeteksi oleh pemilik mobil karena minimnya pengetahuan tentang mesin mobil membuat para pengguna mobil menjadi bingung dan panik serta tidak bisa menangani mobilnya untuk sementara waktu sebelum mobil yang rusak dibawa ke bengkel terdekat. Oleh karena itu dibutuhkan suatu aplikasi yaitu sistem pakar yang merupakan suatu program komputer berbasis pengetahuan dari seorang pakar yang dapat membantu mengurangi resiko kerusakan mobil dengan mengetahui gejala yang terjadi. Salah satu metode sistem pakar yaitu Backward Chaining dimana sudah diketahui permasalahannya kemudian dicari penyebabnya. Untuk mencari rute jalan terpendek yang dibutuhkan oleh seorang user menuju lokasi bengkel maka diperlukan suatu algoritma, salah satunya yaitu algoritma Floyd Warshall yang dapat membandingkan kemungkinan lintasan pada graf untuk setiap sisi dari semua simpul. Sehingga melalui aplikasi ini dapat mempermudah, membantu dan diketahui penyebab-penyebab kerusakan yang terjadi pada mobil dan rute terpendek menuju ke lokasi bengkel.

Kata Kunci : sistem pakar, user, metode backward chaining, algoritma floyd warshall

Abstract

Damage usually occurs suddenly and cannot be detected directly by the car owners because of lack of knowledge about the car engine makes the car users become confused, panic and could not handle the car for a moment before a damaged car was taken to the nearest service station. Therefore, we need an application called expert system, a computer program based on the knowledge of an expert which can help to reduce the risk of damage to the car by knowing the indication of error that was occurred. One of the method is Backward Chaining expert system which can detect the problem and causes. To find the nearest service station location for user by a user requires an algorithm, one of the algorithm was called Floyd Warshall the algorithm system which could compared all the nodes of the graph. Thus, perhaps this application could help the user to detect the causes of damage on the car easier and find the nearest service station.

Keywords : expert system, user, backward chaining method, floyd warshall algorithm

1. PENDAHULUAN

Pada era teknologi saat ini tumbuh dengan sangat pesat terutama perkembangan teknologi dibidang *smartphone* yang rata-rata berbasis Android. Hal ini dibuktikan berdasarkan data dari International Data

Corporation (IDC) bahwa Android pada akhir tahun 2014 mendominasi pasar *smartphone* dunia dengan 76.6% [1]. Pembangunan sistem menggunakan Android

diharapkan dapat membantu dalam menyelesaikan suatu permasalahan.

Kerusakan yang biasa terjadi secara tiba-tiba dan tidak dapat langsung terdeteksi oleh pemilik mobil karena minimnya pengetahuan tentang mesin mobil membuat para pengguna mobil menjadi bingung dan panik serta tidak bisa menangani mobilnya untuk sementara waktu sebelum mobil yang rusak dibawa ke bengkel terdekat. Untuk mengetahui gejala-gejala kerusakan yang terjadi dan pencarian lokasi bengkel maka dibuatlah suatu sistem yaitu sistem pakar diagnosa kerusakan mobil dengan metode *Backward Chaining* dan algoritma pencarian terdekat yaitu *Floyd Warshall*.

Sistem pakar adalah suatu program komputer berbasis pengetahuan yang berusaha seorang pakar ke dalam komputer, agar komputer dapat menyelesaikan masalah yang biasa dilakukan oleh seorang pakar [3]. *Backward Chaining* dipilih karena metode pengambilan dimulai dari tujuan (*goal driven*) selanjutnya dicari aturan yang memiliki tujuan tersebut untuk kesimpulannya [4]. Algoritma *Floyd Warshall* adalah algoritma pencarian terdekat yang membandingkan kemungkinan lintasan pada graf untuk setiap sisi dari semua simpul. Hal tersebut bisa terjadi karena adanya perkiraan pengambilan keputusan (pemilihan jalur terpendek) pada setiap tahap antara dua simpul, hingga perkiraan tersebut diketahui sebagai nilai optimal [7].

2. METODE

2.1 Jenis Penelitian

Penelitian ini adalah penelitian yang berjenis eksperimental. Penelitian eksperimental adalah penelitian dengan mencatat langsung hasil pengujian atau percobaannya dalam pengumpulan data. Untuk mendapatkan kesimpulan dilakukan pengujian secara langsung, baik mengecek kerusakan yang terjadi dengan metode

Backward Chaining dan melakukan pengujian pada algoritma *Floyd Warshall* untuk pencarian rute terpendek.

2.2 Sumber Data

Penelitian ini menggunakan data *rules* kerusakan-kerusakan pada mobil dan data bengkel dealer Toyota di Indonesia yang terdiri dari nama bengkel, alamat bengkel, nomor telepon.

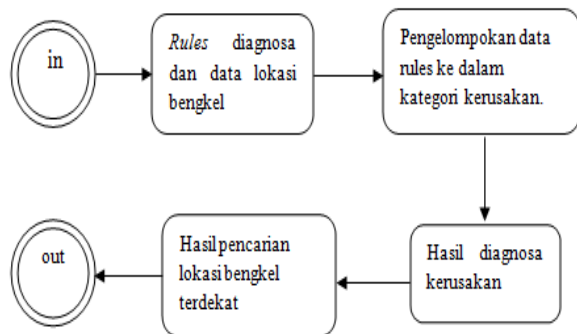
2.3 Analisis Data

Memasukan *rules* yang berisi aturan-aturan dalam menghasilkan diagnosa kerusakan mobil dengan menggunakan metode *Backward Chaining*. Melakukan diagnosa sesuai *rules* dengan menjawab pertanyaan-pertanyaan dimana setiap pertanyaan akan menuju ke diagnosa selanjutnya hingga pertanyaan terakhir dan kemudian akan menampilkan hasil kesimpulan diagnosa kerusakan sementara pada mobil. Memasukan data lokasi-lokasi bengkel ke dalam SQLite yang kemudian memilih nama bengkel yang ingin dituju dan selanjutnya akan diproses menggunakan algoritma *Floyd Warshall* untuk menghasilkan hasil pencarian rute terdekat.

2.4 Metode Yang Diusulkan

Metode yang diusulkan oleh dalam melakukan penelitian adalah menggunakan metode *Backward Chaining* untuk diagnosa kerusakan mobil dan algoritma *Floyd Warshall* untuk pencarian lokasi bengkel terdekat.

Dalam melakukan penelitian ini, prosedur penyelesaian yang nantinya akan dilaksanakan supaya proses penelitian berjalan sesuai dengan tujuan awal dari penelitian yaitu sebagai berikut :



Gambar 1. Prosedur Penyelesaian

Dimasukan *rules* diagnosa rusakand-kerusakan yang ada pada mobil ke dalam Eclipse dengan menggunakan metode *Backward Chaining* dan data lokasi-lokasi bengkel ke dalam SQLite menggunakan algoritma *Floyd Warshall*. *Rules* diagnosa kerusakan dikelompokkan ke beberapa kategori jenis kerusakan yang dialami pada mobil. Setelah memilih kategori jenis kerusakan maka pengguna akan memilih dua kemungkinan yaitu “ya” dan “tidak” dimana setiap dua pilihan tersebut akan menghasilkan diagnosa yang berbeda. Jika hasil diagnosa kerusakan mobil telah diketahui, maka pengguna bisa langsung memilih lokasi bengkel yang ingin dikunjungi.

3. HASIL DAN PEMBAHASAN

Pada penelitian ini telah ditetapkan kaidah untuk mendapatkan kesimpulan berdasarkan fakta dan penghitungan dengan algoritma *Floyd Warshall* untuk medapatkan hasil rute terpendek menuju lokasi bengkel.

3.1 Kaidah Diagnosa Kerusakan Mobil

Sistem pakar memiliki kaidah yang telah ditentukan sebelumnya, berikut ini adalah contoh aturan atau kaidah pada kerusakan mobil :

IF Apakah AC kurang dingin

AND Apakah muncul bau kurang sedap

AND Cek kondisi penyaring udara apakah kotor, jika Ya

THEN Bersihkan penyaring udara hingga bersih.

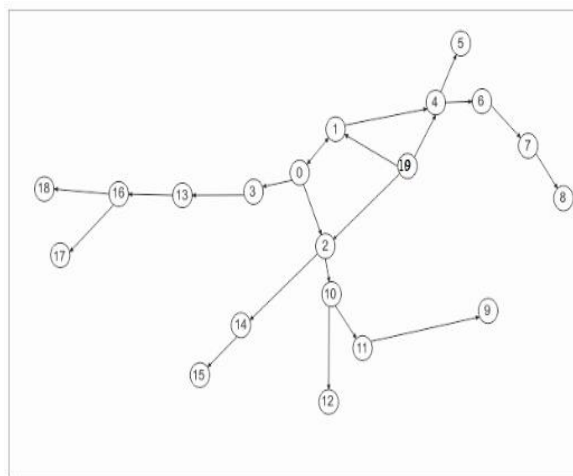
3.2 Perhitungan Algoritma Floyd Warshall

Berikut ini adalah dasar Algoritma *Floyd Warshall* [7]:

```

Procedure Floyd Warshall()
  for k := 1 to n
    for i := 1 to n
      for j := 1 to n
        path[i][j] = min (path [i][j],
        path[i][k] + path [k][j] ;
  
```

kemudian gambar dibawah ini adalah salah satu contoh perhitungan menggunakan algoritma *Floyd Warshall* untuk pencarian rute terpendek :



Gambar 2. Graph Rute Lokasi Bengkel

Dari gambar graph rute lokasi bengkel di atas diketahui terdapat 18 titik atau *node* yang saling berhubungan satu sama lain kemudian langkah selanjutnya adalah membuat tabel data lokasi bengkel seperti berikut :

Tabel 1. Data Jalur Lokasi Bengkel

| No | Kode | Titik Awal | Titik Tujuan | Bobot (Km) |
|-----|------|------------|--------------|------------|
| 1. | A | 0 | 1 | 1.23 |
| 2. | B | 0 | 2 | 8.77 |
| 3. | C | 0 | 3 | 3.06 |
| 4. | D | 1 | 4 | 34.68 |
| 5. | E | 4 | 5 | 36.18 |
| 6. | F | 4 | 6 | 27.21 |
| 7. | G | 6 | 7 | 149.61 |
| 8. | H | 7 | 8 | 102.26 |
| 9. | I | 2 | 10 | 37.47 |
| 10. | J | 10 | 11 | 56.41 |
| 11. | K | 2 | 14 | 189.94 |
| 12. | L | 10 | 12 | 85.16 |
| 13. | M | 11 | 9 | 181.19 |
| 14. | N | 3 | 13 | 157.55 |
| 15. | O | 14 | 15 | 49.49 |
| 16. | P | 13 | 16 | 74.69 |
| 17. | Q | 16 | 17 | 153.87 |
| 18. | R | 16 | 18 | 221.5 |
| 19. | S | 19 | 1 | 5.36 |
| 20. | T | 19 | 2 | 8.77 |

Selanjutnya adalah melakukan langkah pencarian jalur terpendek pada masing-masing titik, dalam hal ini mengambil contoh dari titik awal adalah 0 (X) dan titik tujuan adalah 19 (S).

Tabel 2. Tabel Matriks Iterasi ke 1

| Node | X | A | B | S |
|------|---|------|------|---|
| X | 0 | 1.23 | 8.77 | ~ |

| | | | | |
|---|------|------|------|------|
| A | 1.23 | 0 | ~ | 5.36 |
| B | 8.77 | ~ | 0 | 8.77 |
| S | ~ | 5.36 | 8.77 | 0 |

Dari tabel di atas diketahui titik awal adalah X dengan titik akhir tujuan adalah S. Terdapat 1 titik yang tidak dapat dilalui oleh titik A secara langsung yaitu titik B dan sebaliknya.

Tabel 3. Tabel Matriks Iterasi ke 2

| Node | X | A | B | S |
|------|------|------|------|------|
| X | 0 | 1.23 | 8.77 | ~ |
| A | 1.23 | 0 | 10 | 5.36 |
| B | 8.77 | 10 | 0 | 8.77 |
| S | ~ | 5.36 | 8.77 | 0 |

Berdasarkan tabel tersebut, terdapat dua jalur antara A ke B yaitu :

Jalur 1 : A – X – B = 10

Jalur 2 : A – S – B = 14,31 km

Dari kedua jalur tersebut maka diambil jumlah bobot yang lebih kecil yaitu jalur 1 dengan bobot 10 km.

Tabel 4. Tabel Matriks Iterasi ke 3

| Node | X | A | B | S |
|------|------|------|------|------|
| X | 0 | 1.23 | 8.77 | 6.59 |
| A | 1.23 | 0 | 10 | 5.36 |
| B | 8.77 | 10 | 0 | 8.77 |
| S | 6.59 | 5.36 | 8.77 | 0 |

Berdasarkan tabel tersebut, terdapat dua jalur antara X ke S yaitu :

Jalur 1 : X – B – S = 17.54 km

Jalur 2 : X – A – S = 6.59 km

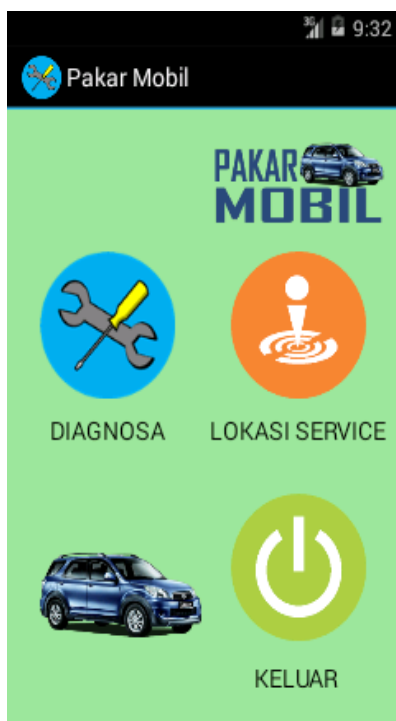
Dari kedua jalur tersebut maka diambil jumlah bobot yang lebih kecil yaitu jalur 1 dengan bobot 6.59 km.

3.3 Tampilan Aplikasi

Implementasi tampilan atau *user interface* merupakan tahap akhir dimana sistem diterapkan secara nyata agar dapat dikelola oleh pengguna. Berikut adalah implementasi *user interface* pada aplikasi sistem pakar diagnosa kerusakan mobil dengan metode *backward chaining* dan pencarian rute bengkel terdekat menggunakan algoritma *floyd warshall*.

1. Tampilan Halaman Utama

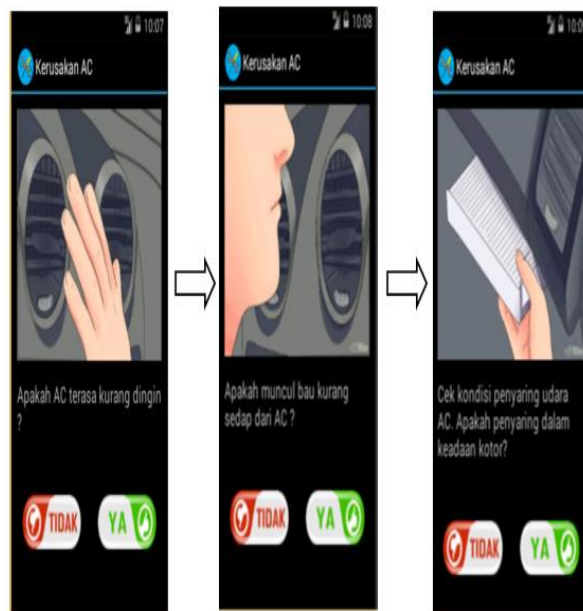
Halaman utama ini merupakan tampilan awal pada saat aplikasi dibuka. Halaman utama ini berisi tiga buah menu yaitu menu diagnosa, menu lokasi service dan menu keluar.



Gambar 3. Tampilan Halaman Utama

2. Tampilan Proses Metode *Backward Chaining*

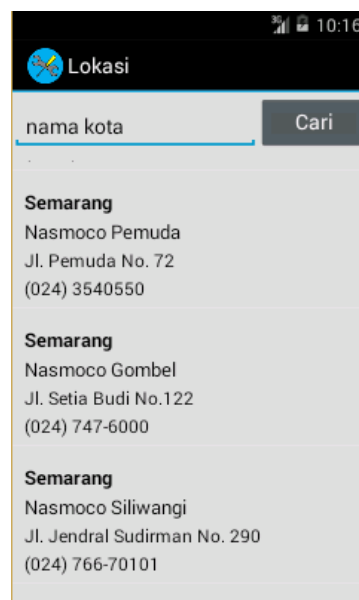
Berikut adalah salah satu contoh tampilan menu diagnosa misalnya pada kategori kerusakan AC dimana sistem akan menampilkan pertanyaan-pertanyaan tentang kerusakan yang terjadi menggunakan metode *backward chaining* dan user akan menjawab “YA” atau “TIDAK”



Gambar 4. Tampilan Menu Diagnosa

3. Tampilan Menu Lokasi Bengkel

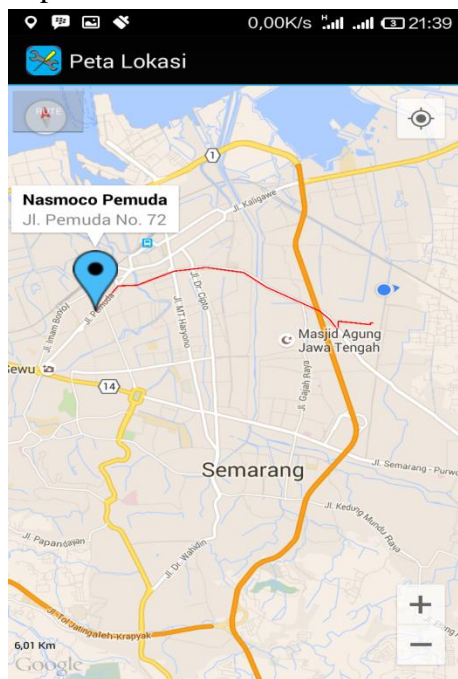
Tampilan Menu Lokasi Bengkel dimana sistem menampilkan list daftar nama-nama bengkel berikut dengan kota, alamat, nomor telepon dan memberikan input kepada user untuk melakukan searching menggunakan button berdasarkan nama kota yang dapat dilihat di bawah berikut :



Gambar 5. Tampilan Menu Lokasi Bengkel

5. Tampilan Peta

Setelah *user* memilih lokasi bengkel yang ingin dituju, kemudian sistem akan menampilkan peta yang berisi posisi user, lokasi bengkel, rute terdekat berupa jalur berwarna merah dan besar jarak yang ditempuh sedangkan untuk rute terdekat jalur berwarna merah tersebut diambil berdasarkan letak antara posisi terdekat user dengan node awal bengkel yang dapat dilihat pada Gambar 6 berikut ini :



Gambar 6. Tampilan Peta dan Rute Terdekat

4. KESIMPULAN DAN SARAN

Berdasarkan dari hasil analisis, perancangan dan implementasi yang telah dibahas pada bab-bab sebelumnya, maka penulis dapat menarik beberapa kesimpulan yaitu :

1. Metode *Backward Chaining* adalah salah satu dari metode yang digunakan dalam sistem pakar dimana metode *Backward Chaining* ini akan mencari tahu penyebab-penyebab kerusakan yang telah terjadi pada mobil.

2. Algoritma *Floyd Warshall* adalah salah satu dari sekian banyak dari algoritma pencarian rute terdekat (*shortest path search*). Implementasi dari algoritma *Floyd Warshall* ini akan menghasilkan rute terpendek menuju bengkel yang ditunjukkan berupa jalur garis berwarna merah.
3. Aplikasi ini memberikan informasi lengkap tentang data bengkel berupa nama, alamat, nomor telepon dan kota.
4. Melalui bantuan google maps dapat diketahui lokasi bengkel, lokasi *user* dan besar jarak dengan satuan kilometer antara *user* dengan bengkel yang ingin dituju.
5. Berdasarkan hasil pengujian *black box* yang valid terhadap 10 kasus uji yang berbeda-beda maka dapat disimpulkan aplikasi ini berjalan dengan baik.

Berikut merupakan beberapa hal yang perlu dilakukan untuk melakukan penelitian lebih lanjut yaitu :

1. Menambahkan lebih banyak lagi lokasi service bengkel Toyota yang ada di seluruh Indonesia.
2. Menambahkan lebih banyak lagi aturan diagnosa kerusakan yang lebih detail dan jelas.
3. Sistem belum bisa menampilkan jalur secara langsung dari posisi user ke bengkel untuk itu penelitian selanjutnya perlu ditambahkan.
4. Menambahkan menu update untuk misalnya menambahkan data lokasi bengkel dan aturan kerusakan sehingga dapat dilakukan secara langsung melalui handphone *user*.

DAFTAR PUSTAKA

- [1] <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>, diakses pada tanggal 3 April 2015.

- [2] <http://www.toyotaku.com/?p=1710>, diakses pada tanggal 3 April 2015.
- [3] Minarni and Rahmat Hidayat, "Rancang Bangun aplikasi Sistem Pakar Untuk Kerusakan Komputer dengan Metode *Backward Chaining*," *Jurnal TEKNOIF*, Vol. 1, No.1, April 2013.
- [4] Listiano. Nur Arif Prabowo, Pinandita. Tito, and Suwarsito, "Sistem Pakar Untuk Mendiagnosa dan Menanggulangi Penyakit Pada Ikan Lele Dumbo (*Clarias Gariepinus*) Menggunakan Metode *Backward Chaining*," *Simposium Nasional Teknologi Terapan (SNTT)*, 2013.
- [5] Setiawan Anton Honggowibowo, "Sistem Pakar Diagnosa Penyakit Tanaman Padi Berbasis Web dengan *Forward* dan *Backward Chaining*," *TELKOMNIKA*, Vol. 7, No. 3, Desember 2009.
- [6] T. Sutojo, Edy Mulyanto and Vincent Suhartono, *Kecerdasan Buatan*. Semarang, Indonesia : Andi, 2010.
- [7] Irfan Ardiansyah and Dimara Kusuma Hakim, "Rancang Bangun Aplikasi untuk Menentukan Jalur Terpendek Menggunakan Algoritma Floyd di Lokasi Wisata Purbalingga," *JUITA*, Vol. II, No. 2, Nopember 2012.
- [8] Purwananto. Yudhi, Purwitasari. Diana, and Wahyu. Agung Wibowo, "Implementasi dan Analisis Algoritma Pencarian Rute Terpendek di Kota Surabaya," *Jurnal Penelitian dan Pengembangan TELEKOMUNIKASI*, Vol. 10, No. 2, Desember 2005.
- [9] Rudi Y. Kriswanto, Kristoforus R. Jawa Bendi, and Aliyanto Arif, "Penentuan Jarak Terpendek Rute Transmisi dengan Algoritma *Floyd-Warshall*," *SEMANTIK*, November 2014.
- [10] Iftadi. Irwan, Ahmad. Wakhid Jauhari, and Nugroho. Beny, "Perancangan Peta Evakuasi Menggunakan Algoritma Floyd-Warshall untuk Penentuan Lintasan Terpendek : Studi kasus," *Performa*, Vol. 10, No 2, pp. 95-104, 2011.
- [11] Pressman, Roger S, *Software Engineering - A Practitioner's Approach*, 7th ed. New York, United States of America : McGraw-Hill, 2009.
- [12] Pressman, Roger S, *Software Engineering - A Practitioner's Approach*, 6th ed. New York, United States of America: McGraw-Hill, 2005.
- [13] J. Stuart Russell and Peter Norvig, *Artificial Intelligence – A Modern Approach*, 3rd ed. Englewood Cliffs, New Jersey : Prentice Hall, 2009.
- [14] I Ketut Ari Wiwekanada. 2012. *Perancangan dan Implementasi Sistem Interaktif Jurnal Pada Sistem Akuntansi (AISO) PT. Dimata Sora Jayate*. Universitas Udayana.
- [15] Artina, Nyimas. 2006. *Penerapan Analisis Kebutuhan Metode Use Case Pada Metode Pengembangan Terstruktur*. STMIK MDP Palembang.
- [16] Anardani, Sri. 2012. *Analisis dan Perancangan Sistem Informasi Berbasis ERP (Enterprise Resource Planning) Di IKIP PGRI Madiun*.