



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1 Knapsack

Istilah *knapsack* dikenal karena karya seorang matematikawan Tobias Dantzig.

2.1.1 Definisi Knapsack

Knapsack adalah tas atau karung yang digunakan untuk memasukkan sesuatu. Tapi tidak semua barang bisa ditampung ke dalam karung tersebut. Karung tersebut hanya dapat menyimpan beberapa objek dengan total ukurannya (weight) lebih kecil atau sama dengan ukuran kapasitas karung (Roy Sahala Aruan, 2010). *Knapsack problem* memiliki tiga jenis persoalan, yaitu:

a. *Knapsack 0/1*

Sesuatu yang dimasukkan ke dalam karung dimensinya harus dimasukkan semua atau tidak sama sekali.

b. *Knapsack Bounded*

Sesuatu yang dimasukkan ke dalam karung dimensinya bisa dimasukkan sebagian atau seluruhnya.

c. *Knapsack Unbounded*

Untuk menyelesaikan masalah knapsack ini digunakan Algoritma *greedy*. Algoritma ini memecahkan permasalahan langkah per langkah, yaitu:

- a. Mengambil pilihan terbaik yang bisa diperoleh saat itu juga tanpa memperhatikan konsekuensi kedepan (prinsip “*take what you can get now!*”).

- b. Berharap bahwa dengan memilih optimum lokal pada setiap langkah akan berakhir dengan optimum global.

Pada *Greedy Algorithm* ada beberapa strategi yang digunakan untuk memilih objek yang akan dimasukkan kedalam knapsack (Roy Sahala Aruan,2010):

- a. *Greedy by profit*

Pada setiap langkah, *knapsack* diisi dengan objek yang mempunyai keuntungan terbesar. Strategi ini mencoba memaksimalkan keuntungan dengan memilih objek yang paling menguntungkan terlebih dahulu.

- b. *Greedy by weight*(algoritma yang digunakan)

Pada setiap langkah,*knapsack* diisi dengan objek yang mempunyai berat paling ringan. Strategi ini mencoba memaksimalkan keuntungan dengan memasukkan sebanyak mungkin objek ke dalam knapsack

- c. *Greedy by density*

Pada setiap langkah, *knapsack* diisi dengan objek yang mempunyai densitas, p_i/w_i terbesar. Strategi ini mencoba memaksimalkan keuntungan dengan memilih objek yang mempunyai keuntungan per unit berat terbesar.

2.1.2 Masalah Knapsack

Masalah *knapsack* merupakan sebuah persoalan yang menarik. Dalam dunia nyata permasalahan *knapsack* ini sering sekali digunakan terutama pada bidang (jasa) pengangkutan barang (seperti pengangkutan peti kemas dalam sebuah kapal). Dalam usaha tersebut, diinginkan suatu keuntungan yang maksimal untuk mengangkut barang yang ada dengan tidak melebihi batas kapasitas yang ada. Berdasarkan persoalan tersebut, diharapkan ada suatu solusi yang secara otomatis dalam mengatasi persoalan itu. *Problem Knapsack* adalah permasalahan

optimasi kombinatorial, dimana kita harus mencari solusi terbaik dari banyak kemungkinan yang dihasilkan(Roy Sahala Aruan,2010).

2.1.3 Penyelesaian masalah Knapsack

Berikut adalah beberapa cara penyelesaian *knapsack*(Roy Sahala Aruan,2010) :

1. *Brute Force*

Brute force adalah pendekatan straightforward untuk menyelesaikan masalah,umumnya sangat bergantung pada pernyataan masalah dan definisi dari konsep. Jika terdapat n item untuk dipilih, maka akan ada 2^n kemungkinan kombinasi dari *item-item* tersebut untuk ditempatkan di *Knapsack*. Sebuah item dapat terpilih atau tidak dalam kombinasi tersebut. Angka 0 dan 1 akan dibangkitkan sepanjang n . Jika i menunjukkan 0 maka item tersebut tidak terpilih dan jika 1 maka item tersebut dipilih.

2. Algoritma Genetika

Algoritma genetik merupakan algoritma komputer yang mencari suatu solusi-solusi baik dalam permasalahan yang memiliki sejumlah besar kemungkinan pemecahanyang ada. Semua algoritma-algoritma genetik dimulai dengan kumpulan solusi (yang diwakili oleh kromosom) yang biasa disebut populasi. Suatu populasi baru diciptakan dari solusi-solusi yang ada dalam suatu populasi tua diharapkan dapat menjadi suatu populasi lebih baik. Solusi-solusi yang telah dipilih dalam membentuk solusi baru(anak/offspings) akan diseleksi menurut fitness mereka. Semakin solusi-solusinya tersebut cocok maka akan lebih banyak kesempatan mereka dalam produksi kembali. Proses ini diulangi sampai kondisi yang diinginkan didapat.Kebanyakan algoritma genetik berdasarkan atas element-element berikut:

“populasi dari kromosom, pemilihan berdasarkan *fitness*, penyilangan dalam mendapatkan *offspings* baru, dan mutasi acak dalam *offspings* baru”.

3. Algoritma *Greedy*

Teknik pemrograman dengan menggunakan *Greedy* sering digunakan untuk permasalahan optimasi. Secara umum teknik ini menggunakan heuristic untuk mencari solusi suboptimum sehingga diharapkan solusi optimum. Strategi greedy yang dapat diterapkan pada 0/1 Knapsack Problem :

1. Pilih item yang memiliki nilai maksimum dari item-item yang tersedia, hal ini akan menambah nilai dari *Knapsack* secara cepat.
2. Pilih item yang memiliki bobot minimum dari item-item yang ada sehingga kapasitas terisi secara perlahan dan dapat memuat lebih banyak item.
3. Pilih item yang memiliki nilai tinggi untuk bobot/berat

Setelah tiga strategi tersebut diterapkan dan diuji, maka didapat hasil terbaik dari aturan ketiga, yaitu memilih item bernilai tinggi dari rasio bobot terhadap berat.

2.1.4 Contoh Persoalan

Contoh 1 (Masalah Penukaran uang):

Persoalan: Diberikan uang senilai A . Tukar A dengan koin-koin uang yang ada. Berapa jumlah minimum koin yang diperlukan untuk penukaran tersebut?

(Munir Rinaldi, 2004)

Contoh: tersedia koin-koin 1, 5, 10, dan 25

Uang senilai 32 dapat ditukar dengan cara berikut:

$$32 = 1 + 1 + \dots + 1 \quad (32 \text{ koin})$$

$$32 = 5 + 5 + 5 + 5 + 10 + 1 + 1 \quad (7 \text{ koin})$$

$$32 = 10 + 10 + 10 + 1 + 1 \quad (5 \text{ koin})$$

... dan seterusnya

Minimum: $32 = 25 + 5 + 1 + 1$) hanya 4 koin

Strategi *greedy* yang digunakan adalah:

Pada setiap langkah, pilihlah koin dengan nilai sebesar mungkin dari himpunan koin yang tersisa dengan syarat (kendala) tidak melebihi nilai uang yang ditukarkan. (Munir, Rinaldi.2004)

Tinjau masalah menukarkan uang 32 dengan koin 1, 5, 10, dan 25:

Langkah 1: pilih 1 buah koin 25 (Total = 25)

Langkah 2: pilih 1 buah koin 5 (Total = $25 + 5 = 30$)

Langkah 3: pilih 2 buah koin 1 (Total = $25+5+1+1= 32$)

Solusi: Jumlah koin minimum = 4 (solusi optimal!)

Pada setiap langkah di atas kita memperoleh optimum lokal, dan pada akhir algoritma kita memperoleh optimum global (yang pada contoh ini merupakan solusi optimum).(Munir Rinaldi,2004)

Algoritma *greedy* memiliki skema umum di antaranya: (Munir Rinaldi,2004)

1. Himpunan kandidat.

Berisi elemen-elemen pembentuk solusi.

2. Himpunan solusi

Berisi kandidat-kandidat yang terpilih sebagai solusi persoalan.

3. Fungsi seleksi (*selection function*)

Memilih kandidat yang paling memungkinkan mencapai solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya.

4. Fungsi kelayakan (*feasible*)

Memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar kendala (*constraints*) yang ada. Kandidat yang layak dimasukkan ke dalam himpunan solusi, sedangkan kandidat yang tidak layak dibuang dan tidak pernah dipertimbangkan lagi.

5. Fungsi obyektif, yaitu fungsi yang memaksimumkan atau meminimumkan nilai solusi (misalnya panjang lintasan, keuntungan, dan lain-lain).

Contoh pada masalah penukaran uang, elemen-elemen algoritma *greedy*-nya adalah: (Munir Rinaldi, 2004)

1. Himpunan kandidat: himpunan koin yang merepresentasikan nilai 1, 5, 10, 25, paling sedikit mengandung satu koin untuk setiap nilai.
2. Himpunan solusi: total nilai koin yang dipilih tepat sama jumlahnya dengan nilai uang yang ditukarkan.
3. Fungsi seleksi: pilihlah koin yang bernilai tertinggi dari himpunan kandidat yang tersisa.
4. Fungsi layak: memeriksa apakah nilai total dari himpunan koin yang dipilih tidak melebihi jumlah uang yang harus dibayar.
5. Fungsi obyektif: jumlah koin yang digunakan minimum.

```

procedure greedy(input C: himpunan_kandidat;
                 output S : himpunan_solusi)
{ menentukan solusi optimum dari persoalan optimasi dengan algoritma
  greedy
  Masukan: himpunan kandidat C
  Keluaran: himpunan solusi S
}
Deklarasi
  x : kandidat;

Algoritma:
  S ← {}           { inisialisasi S dengan kosong }
  while (belum SOLUSI(S)) and (C ≠ {} ) do
    x ← SELEKSI(C); { pilih sebuah kandidat dari C }
    C ← C - {x}    { elemen himpunan kandidat berkurang satu }
    if LAYAK(S ∪ {x}) then
      S ← S ∪ {x}
    Endif
  endwhile
  {SOLUSI(S) sudah diperoleh or C = {} }

```

Gambar 2.1 Contoh potongan *pseudo code* algoritma *greedy*

- Pada akhir setiap lelaran, solusi yang terbentuk adalah optimum lokal.
Pada akhir kalang while-do diperoleh optimum global.

- Namun adakalanya optimum global merupakan solusi *sub-optimum* atau *pseudo-optimum*. Alasan:
 1. algoritma *greedy* tidak beroperasi secara menyeluruh terhadap semua alternatif solusi yang ada (sebagaimana pada metode *exhaustive search*).
 2. pemilihan fungsi SELEKSI: Mungkin saja terdapat beberapa fungsi SELEKSI yang berbeda, sehingga kita harus memilih fungsi yang tepat jika kita ingin algoritma bekerja dengan benar dan menghasilkan solusi yang benar-benar optimum
- Karena itu, pada sebagian masalah algoritma *greedy* tidak selalu berhasil memberikan solusi yang benar-benar optimum.
- Jika jawaban terbaik mutlak (benar-benar optimum) tidak diperlukan, maka algoritma *greedy* sering berguna untuk menghasilkan solusi yang menghampiri (*approximation*) optimum, daripada menggunakan algoritma yang lebih rumit untuk menghasilkan solusi yang eksak.
- Bila algoritma *greedy* optimum, maka keoptimalannya itu dapat dibuktikan secara matematis

- Contoh Soal :

Persoalan: Sebuah *server* (dapat berupa *processor*, pompa, kasir di bank, dll) mempunyai n pelanggan (*customer*, *client*) yang harus dilayani. Waktu pelayanan untuk setiap pelanggan sudah ditetapkan sebelumnya, yaitu pelanggan i membutuhkan waktu t_i . Kita ingin meminimumkan total waktu di dalam sistem, (Munir, Rinaldi.2004)

$$T = \sum_{i=1}^n (\text{waktu di dalam sistem untuk pelanggan } i)$$

Rumus 2.1 Contoh rumus waktu pelayanan

Karena jumlah pelanggan adalah tetap, meminimumkan waktu di dalam sistem ekuivalen dengan meminimumkan waktu rata-rata.

Contoh . Misalkan kita mempunyai tiga pelanggan dengan

$$t_1 = 5, \quad t_2 = 10, \quad t_3 = 3,$$

maka enam urutan pelayanan yang mungkin adalah:

Urutan	T
=====	=====

$$1, 2, 3: 5 + (5 + 10) + (5 + 10 + 3) = 38$$

$$1, 3, 2: 5 + (5 + 3) + (5 + 3 + 10) = 31$$

$$2, 1, 3: 10 + (10 + 5) + (10 + 5 + 3) = 43$$

$$2, 3, 1: 10 + (10 + 3) + (10 + 3 + 5) = 41$$

$$\mathbf{3, 1, 2: 3 + (3 + 5) + (3 + 5 + 10) = 29 \leftarrow (\text{optimal})}$$

$$3, 2, 1: 3 + (3 + 10) + (3 + 10 + 5) = 34$$

Pemecahan Masalah dengan Algoritma Exhaustive Search

- Urutan pelanggan yang dilayani oleh *server* merupakan suatu permutasi

- Jika ada n orang pelanggan, maka terdapat $n!$ urutan pelanggan. Waktu yang dibutuhkan untuk mengevaluasi fungsi obyektif adalah $O(n)$, oleh karena itu kompleksitas algoritma *exhaustive search* untuk masalah ini adalah $O(nn!)$

Pemecahan Masalah dengan Algoritma Greedy (Munir Rinaldi, 2004)

Strategi *greedy* untuk memilih pelanggan berikutnya adalah:

- Pada setiap langkah, masukkan pelanggan yang membutuhkan waktu pelayanan terkecil di antara pelanggan lain yang belum dilayani.
- Agar proses pemilihan pelanggan berikutnya optimal, maka perlu mengurutkan waktu pelayanan seluruh pelanggan dalam urutan yang menaik. Jika waktu pengurutan tidak dihitung, maka kompleksitas algoritma *greedy* untuk masalah minimisasi waktu di dalam sistem adalah $O(n)$.

UMMN

```
procedure PenjadwalanPelanggan(input n:integer)
```

```
{ Mencetak informasi deretan pelanggan yang akan diproses oleh server tunggal
```

```
Masukan: n pelanggan, setiap pelanggan dinomori 1, 2, ..., n
```

```
Keluaran: urutan pelanggan yang dilayani
```

```
}
```

```
Deklarasi
```

```
i : integer
```

```
Algoritma:
```

```
{pelanggan 1, 2, ..., n sudah diurut menaik berdasarkan  $t_i$ }
```

```
for i←1 to n do
```

```
    write('Pelanggan ', i, ' dilayani!')
```

```
endfor
```

Gambar 2.2 *pseudo code* contoh persoalan pelayanan

Pemilihan strategi *greedy* untuk penjadwalan pelanggan akan selalu menghasilkan solusi optimum. Keoptimuman ini dinyatakan dengan Teorema 2.1 berikut:

(Munir, Rinaldi.2004)

Teorema 2.1. Jika $t_1 \leq t_2 \leq \dots \leq t_n$ maka pengurutan $i_j = j$, $1 \leq j \leq n$ meminimumkan

$$T = \sum_{k=1}^n \sum_{j=1}^k t_{i_j}$$

Rumus 2.2 teorema keoptimum pelayanan pelanggan

untuk semua kemungkinan permutasi i_j .

2.2 Algoritma

Menurut Utami dan Sukrisno (2005: 19), algoritma adalah suatu metode atau logika tentang urutan mengerjakan dan memecahkan suatu masalah secara sistematis. Algoritma yang baik akan menghasilkan keluaran yang sesuai dari masukan yang dikehendaki.

2.3 MySql

2.3.1 Definisi MySql

Menurut Kustiyahningsih (2011:145), “MySQL adalah sebuah basis data yang mengandung satu atau jumlah table. Table terdiri atas sejumlah baris dan setiap baris mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau sejumlah tabel”.

Menurut Wahana Komputer (2010:21), MySQL adalah database server open source yang cukup populer keberadaanya. Dengan berbagai keunggulan yang dimiliki, membuat software database ini banyak digunakan oleh praktisi untuk membangun suatu project. Adanya fasilitas API (*Application Programming Interface*) yang dimiliki oleh Mysql, memungkinkan bermacam-macam aplikasi Komputer yang ditulis dengan berbagai bahasa pemrograman dapat mengakses basis data MySQL.

MYSQL termasuk jenis RDBMS (*Relational Database Management System*). Sedangkan RDMS sendiri akan lebih banyak mengenal istilah seperti table, baris dan kolom digunakan dalam perintah-perintah di MYSQL.

2.4 Bengkel

2.4.1 Pengertian Bengkel

Pengertian bengkel adalah suatu tempat dimana dilakukan perbaikan-perbaikan yang bersifat teknis terhadap suatu produk yang dalam konteks materi ini, produk yang dimaksud adalah kendaraan bermotor. Sebetulnya kegiatan perbengkelan adalah bagian dari kegiatan jaringan layanan purna jual yang sekaligus berfungsi mendukung pemasaran produk yang dijual (yang dalam hal ini adalah kendaraan bermotor). Dalam kenyataannya layanan tidak hanya diberikan kepada kendaraan, tetapi diberikan pula kepada manusianya yaitu pemilik kendaraan itu sendiri, sehingga mutu pelayanan bagi keduanya harus menjadi perhatian yang serius. (Permana Tatang,2010)

Materi yang diberikan umumnya berfokus kepada perbengkelan kendaraan beroda empat atau lebih, namun masih memungkinkan diaplikasikan untuk kendaraan beroda dua atau lainnya yang juga masih tergolong otomotif. Ada beberapa jenis dan status bengkel yang dapat diterangkan sebagai berikut : (Permana Tatang,2010)

1. Bengkel Bebas (*Independent Work Shop*)

Bengkel ini berdiri sendiri, tidak terikat dan tidak memwakili merek tertentu sehingga kebijakan-kebijakan dapat diambil sendiri sepanjang tidak merugikan bengkel itu sendiri sebagai perusahaan atau sepanjang tidak merusak nama baik perusahaan pemegang merek.

2. Bengkel Perwakilan (*Authorized Work Shop*)

Bengkel ini masih mirip dengan bengkel tersebut diatas, yaitu berdiri sendiri tapi ada merek yang diwakilinya melalui surat penunjukan dari pemegang merek. Kebijakan-kebijakan yang diambil disesuaikan dengan perusahaan yang menunjuknya dan sekaligus masuk kedalam bagian dari layanan purna jual merek yang bersangkutan. Jenis bengkel ini memungkinkan untuk menerima kemudahan-kemudahan dari perusahaan yang menunjuknya. Kemudahan-kemudahan tersebut bisa bersifat bantuan teknis, permodalan, peralatan atau jenis kemudahan lainnya tergantung dari kebijakan perusahaan yang menunjuknya dan kesepakatan/perjanjian yang dibuat diantara keduanya.

3. Bengkel *Dealer* (*Dealer Work Shop*)

Bengkel ini merupakan bagian atau sub bagian operasional dari dealer atau ATPM (Agen Tunggal Pemegang Merek) sebagai unit layanan purna jual untuk mendukung sistem pemasaran. Kebijakan-kebijakan yang dibuat sepenuhnya tergantung dan tunduk kepada perusahaan/*dealer* yang bersangkutan.

Bagaimanapun sebuah bengkel adalah sebuah bentuk usaha sehingga secara operasional harus menguntungkan (*Profitable*) dan layak (*Feasible*). Oleh karenanya seluruh kegiatan harus berorientasi kepada perolehan laba. Namun demikian perlu diingatkan bahwa laba harus diperoleh dengan cara yang wajar dan sehat sebab jika tidak demikian justru akan sangat merugikan bengkel itu sendiri karena akan segera ditinggalkan oleh para pelanggannya. Bukankah sumber keuangan bengkel adalah pelanggannya?. (Permana Tatang,2010)

Adapun ruang lingkup pekerjaan bengkel diantaranya adalah sebagai berikut :

1. Layanan cepat (*Quick service*) dapat berupa, pekerjaan *tune-up*, mengganti minyak pelumas, mencuci dan lain-lain.

2. Perbaikan umum (*General repair*) yang berupa perbaikan *engine*, transmisi, differensial, penyetelan geometrid dan balancing roda.

3. Perbaikan elektrik (*Electrical repair*) yang berupa perbaikan system pengapian, starter, pengisian, system penerangan dan instrument.

4. Perbaikan system pendingin ruangan (*Car cooler and Air conditioning*)

5. *Over haul and reconditioning*

6. Perbaikan mesin seperti boring, honing, bubut rem, skir katup dan lain-lain

7. Perbaikan body kendaraan dan cat

8. Perbaikan yang bersifat *fashion* (salon)

9. Pemasangan *accessories* dan *optional parts*

10. Pekerjaan lainnya

Beberapa jenis pekerjaan yang dapat dilakukan sangat tergantung kepada skala bengkel yang harus dipertimbangkan dari bayak hal, misalnya permodalan, jumlah pelanggan, lokasi bengkel, segmen pasar yang diharapkan dan lain-lain.

(Permana Tatang,2010)

2.4.2 Fungsi Bengkel

Secara umum fungsi bengkel adalah melayani keperluan teknis dari para pelanggannya. Ini berarti bahwa perbaikan kendaraan adalah tugas sebuah bengkel dan hanya berlangsung jika pelanggan menemui kesulitan dengan

kendaraannya. Untuk itu system dan administrasi bengkel diarahkan kepada organisasi dan fasilitas yang dapat memperlancar pekerjaan-pekerjaan teknis dibengkel secara internal. Bengkel tradisional pada umumnya menunggu para pelanggan datang, usaha untuk mendatangkan pelanggan hamper tidak pernah dilakukan sehingga pengembangan usaha berjalan lambat dan cenderung tidak menunjukkan kemajuan yang berarti. Manajemen bengkel diarahkan hanya untuk membenahan didalam bengkel itu sendiri sambil tidak memperhitungkan dampak sosial, ekonomi, dan politik yang sedang berlangsung atau yang mungkin datang dari luar. (Permana Tatang,2010)

Ciri lain dari bengkel tradisional adalah kondisinya yang kurang teratur dan kelihatan mengabaikan kebersihan, kerapihan dan keserasian. Penempatan peralatan, susunan dan penempatan barang, tata ruang dan fasilitas pendukung tidak nampak tertata dengan baik yang penting adalah seorang teknisi dapat menyelesaikan pekerjaannya secara baik. Apakah ia bekerja efisien dan produktif itu menjadi urusan lain. (Permana Tatang,2010)

Pada masa kini bengkel dituntut untuk mempunyai pola pikir dan konsep operasional yang berbeda. Bengkel yang kumuh dan kotor akan sedikit didatangi pelanggan bahkan mungkin tidak ada pelanggan yang datang. Fasilitas yang sangat terbatas membuat bengkel tersebut tidak credible dan pelanggan enggan datang kepadanya. Fasilitas yang dimaksud tidak sekedar fasilitas teknis bagi bengkel tapi juga fasilitas yang harus tersedia bagi pelanggannya. Mau tidak mau selama kendaraan dikerjakan pelanggan akan menunggu dan tentu tidak ingin kehilangan waktu. Jadi harus dipikirkan bagaimana supaya pelanggan tidak menunggu dan jika harus menunggu maka harus difikirkan agar waktunya tidak

terbuang percuma. Jika pimpinan bengkel jeli sebetulnya hal ini merupakan peluang yang cukup besar untuk membuka bisnis lain dengan pelanggan tersebut misalnya menawarkan *accessories* atau barang-barang lainnya. (Permana Tatang,2010)

Kepala bengkel harus proaktif melakukan pendekatan dengan para pelanggan, tidak cukup diam dimeja kerja dan menandatangani surat-surat yang memerlukan persetujuan atau keputusan darinya. Pimpinan bengkel harus dapat melakukan pendekatan sedemikian rupa sehingga para pelanggan tidak pernah merasa datang ketempat asing baginya, bahkan ia merasa datang ketempat yang layak serasa dirumahnya sendiri dan dilayani oleh kerabatnya sendiri. Disinilah timbulnya kredibilitas seorang pimpina bengkel dan sekaligus pula kredibilitas bengkel itu sendiri. Hubungan dengan pelanggan tidak saja pada saat pelanggan tersebut membutuhkan, namun harus tetap berlangsung disembarang waktu, bahkan mungkin pada saat tertentu yang merupakan kejutan khusus bagi pelanggan yang bersangkutan. (Permana Tatang,2010)

Apa yang menjadi fungsi pokok dalam mengoperasikan sebuah bengkel harus dilihat dulu apa yang dibutuhkan oleh pelanggannya, kemudian bengkel segera melakukan usaha-usaha dan menyesuaikan diri untuk memenuhi kebutuhan tersebut hal ini disebut manajemen dinamis (*Dynamic Management*) yang berpola fikir *customer focused mindset* sedangkan pola berfikir bengkel tradisional disebut sebagai *self focused mindset*. Pelanggan mengharapkan agar bengkel dapat membantu merawat kendaraan secara baik, membantu melakukan tindakan preventif, memperbaiki kerusakan yang mungkin terjadi, menekan biaya perawatan dimana pelanggan ingin agar kendaraan selalu dalam keadaan prima.

Seringkali pelanggan memerlukan nasihat dan saran untuk perawatan kendaraan bahkan tidak jarang pelanggan ingin agar kendaraannya tampil beda, eksklusif sehingga menaikkan nilai harga dirinya atau kebangganya. (Permana Tatang,2010)

Bengkel harus dapat mengarahkan pelanggan agar memelihara kendaraan secara berkesinambungan bahkan kalau perlu sampai pada saat pelanggan tersebut ingin mengganti dengan kendaraan lainnya yang menurut pendapatnya lebih baik atau lebih cocok baginya. Selain itu bengkel perlu membimbing pelanggan untuk melakukan perawatan berkala sebagai dampaknya kendaraan akan terawat baik dan pelanggannya sendiri terbentuk menjadi pelanggan setia (*Loyal Customer*). (Permana Tatang,2010)

Kunci pokok keberhasilan sebuah bengkel dapat dicapai jika bengkel tersebut mampu memberikan layanan total kepada pelanggannya sebagai berikut :

- a. Dapat memberikan pelayanan, perbaikan, perawatan bermutu tinggi kepada kendaraan pelanggannya
- b. Dapat mengatur waktu sehingga kendaraan dapat diserahkan sesuai dengan waktu yang dijanjikan
- c. Menentukan biaya kerja dan harga suku cadang yang wajar
- d. Melayani pelanggannya dengan santu dan penuh keakraban
- e. Melakukan pekerjaan yang efisien, akurat dan administrasi yang rapih.

Dari uraian di atas dapat disimpulkan bahwa fungsi bengkel otomotif terdiri dari : (Permana Tatang,2010)

- a. Penjamin purna jual
- b. Promosi dan *image*
- c. Bisnis (dagang/usaha)
- d. Lapangan kerja
- e. Pendidikan dan latihan/diklat

Fungsi bengkel tersebut berhubungan dengan tanggung jawab bengkel otomotif itu sendiri yang meliputi : (Permana Tatang,2010)

1. Melakukan perawatan berkala
2. Menjamin keamanan pengendara
3. Menjamin keselamatan peserta lalu lintas jalan darat
4. Melestarikan lingkungan
5. Meningkatkan efisiensi pemakaian bahan bakar
6. Menjamin kelancaran lalu lintas jalan darat

UMMN