

Pengembangan *Case Based Reasoning* pada Aplikasi Pemesanan Kain Berdasarkan Studi Kasus pada CV. Mitra KH Bandung

Hapnes Toba¹, Sylvia Tanadi²

Program Studi D3 Teknik Informatika
Fakultas Teknologi Informasi Universitas Kristen Maranatha
Jl. Prof. Drg. Suria Sumantri No. 65 Bandung 40164
email : hapnes.toba@eng.maranatha.edu ¹⁾, sylviatanadi@gmail.com ²⁾

Abstract

The main aim of this application is to assist customers who wants to place a sales order via a web based system without going to the factory. Customer who wants to place an order can access the web application via a computer or a mobile device. This application uses Case Based Reasoning method and Weighted Euclidean Distance algorithm. The purpose of using Case Based Reasoning and Weighted Euclidean Distance algorithm is that the system can give how long an order can be done according to some similar cases that has been stored in the history database.

Keywords : Case Based Reasoning, Weighted Euclidean Distance, Program Evaluation and Review Technique (PERT).

1. Pendahuluan

CV Mitra KH merupakan sebuah perusahaan yang bergerak dalam bidang pembuatan produk garmen (kain). Perusahaan ini memerlukan sebuah aplikasi yang memudahkan pemesanan kain bagi para pelanggannya. Melalui aplikasi yang dimaksudkan, pelanggan akan dapat melihat perkiraan berapa lama sebuah pemesanan akan dapat dilakukan, dengan demikian dapat pula "mendorong" bagian produksi untuk memenuhi target. Pengajuan pemesanan biasanya dilakukan oleh pelanggan yang sudah lama menjadi pelanggan. Untuk para pelanggan yang demikian ini sudah tersimpan informasi historis pemesanan dan bagaimanapun produksi kain dibuat di dalam pabrik untuk pemesanan tersebut. Informasi historis ini dapat pula digunakan untuk memprediksi pesanan yang mirip jika dilakukan oleh pelanggan baru.

Pada aplikasi pemesanan kain ini akan diterapkan metode *Case Based Reasoning* (CBR) dengan menerapkan algoritma *Weighted Euclidean Distance*. Alasan digunakannya CBR, karena dalam sebuah sistem pemesanan, untuk mempercepat perkiraan lama waktu pemesanan, diperlukan pencocokan antara kasus yang lama dengan kasus yang baru. Apabila terdapat kesamaan antara kasus yang lama dan baru, solusi yang telah didapat dari kasus sebelumnya akan digunakan kembali sebagai solusi bagi kasus yang baru. Apabila dalam melakukan pencocokan, sistem tidak menemukan kasus lama yang sama dengan kasus baru maka sistem akan melakukan pembelajaran untuk dapat menemukan solusi bagi kasus baru tersebut.

2. Metode *Case Based Reasoning*

Case Based Reasoning (CBR) merupakan salah satu metode pemecahan masalah yang dalam mencari solusi dari suatu kasus yang baru, sistem akan melakukan

pencarian terhadap solusi dari kasus lama yang memiliki permasalahan yang sama dan sudah pernah terjadi sebelumnya. [Rus2003]. Metode CBR dikembangkan oleh Roger Schank dan rekannya di Universitas Yale pada awal tahun 1980. Terdapat dua prinsip dasar pada metode CBR, prinsip pertama adalah setiap permasalahan yang sama akan memiliki solusi yang sama pula. Oleh karena itu, solusi dari permasalahan yang sudah pernah terjadi dapat digunakan kembali untuk memecahkan masalah baru dengan permasalahan yang sama dengan masalah yang lama. Prinsip kedua adalah setiap permasalahan dapat terjadi berulang kali. Oleh karena itu, terdapat kemungkinan bahwa masalah yang akan muncul di masa yang akan datang memiliki kesamaan dengan masalah yang pernah terjadi sebelumnya.

Terdapat empat proses yang terjadi pada metode CBR dalam menyelesaikan masalah, yaitu [lihat Gambar 1] :

- *Retrieve*

Pada proses ini sistem akan melakukan identifikasi parameter pencocokan yang dapat dijadikan sebagai acuan lalu melakukan pencarian kasus lama yang memiliki kesamaan dengan kasus baru. Selanjutnya sistem akan melakukan pencocokan parameter antara kasus baru dengan kasus lama dan memilih kasus yang memiliki tingkat kecocokan tertinggi.

- *Reuse*

Pada proses ini sistem akan menggunakan kembali informasi yang berasal dari kasus sebelumnya atau sistem akan melakukan adaptasi terlebih dahulu untuk memecahkan masalah pada kasus yang baru.

- *Revise*

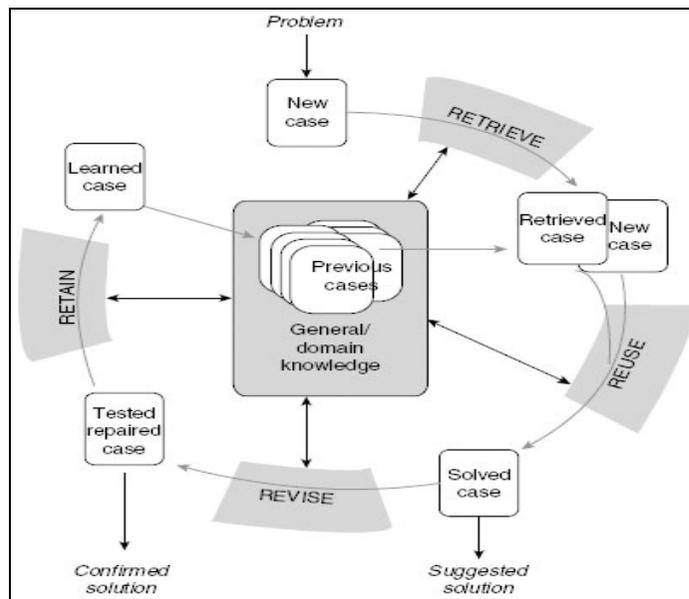
Pada proses ini sistem akan meninjau kembali solusi yang telah didapatkan dari kasus yang lama apakah solusi tersebut akan diterapkan pada kasus yang baru atau solusi tersebut perlu diperbaiki terlebih dahulu.

- *Retain*

Pada proses ini apabila ternyata ditemukan solusi baru yang lebih baik dari solusi yang telah ada sebelumnya maka solusi baru tersebut akan diberi indeks dan disimpan untuk kemudian digunakan kembali pada kasus serupa pada masa yang akan datang.

Penerapan CBR banyak dilakukan untuk memodelkan domain permasalahan yang (bisa merupakan kombinasi) [Wat1994]:

1. Sulit dalam melakukan elisitasi pengetahuan;
2. Sulit menerapkan aturan *if-then*, karena sangat bergantung pada tingkat kepakaran yang spesifik dan berkembang dalam waktu lama;



Gambar 1 Case Based Reasoning Cycle [Aam1994]

3. Jika penerapan aturan dapat dilakukan, maka aturan dapat bertahan lama dan besaran data akan dapat berkembang pesat;
4. Ada kecenderungan sulit dalam pengelolaan data karena perkembangannya yang pesat;
5. Banyak memiliki pengetahuan historis.

Dalam awal perkembangannya konsep CBR banyak diterapkan dalam domain pembuatan rencana perjalanan, misalnya CYRUS [Kol1983], ataupun penyelesaian kasus pengadilan (hukum), seperti pada HYPO [Ash1988].

3. Pemodelan Kasus dan Algoritma *Weighted Euclidean Distance*

Representasi Kasus dan Mekanisme CBR

Hal pertama yang harus dilakukan dalam sebuah aplikasi dengan CBR adalah melakukan pemodelan kasusnya. Untuk aplikasi pemesanan kain di CV Mitra KH, pemodelan kasus yang dapat dilakukan adalah dengan memperhitungkan parameter pemesanan dan parameter produksi untuk identifikasi lama waktu produksi kain sesuai karakteristik kain yang dipesan dan mesin produksinya.

Sebuah kasus akan diidentifikasi dengan: jenis mesin, jenis benang, ketebalan, gramasi, kerapatan, *finishing*, warna, dan kuantitas pemesanan. Jenis mesin merupakan prioritas terbesar dalam memperhitungkan waktu produksi, dan parameter inilah yang akan menjadi parameter penentu dalam indeksasi kasus.

Pemilihan parameter-paramater di atas dapat dijelaskan dengan mekanisme sebagai berikut: pada saat seorang pemesan melakukan pemesanan, akan dimasukkan data: pelanggan, warna kain, ketebalan, gramasi, kerapatan, serta jumlah pemesanan.

Lama waktu pemesanan akan sangat tergantung pada ketersediaan stok di dalam gudang, dan ketersediaan mesin produksi. Setiap mesin memiliki spesialisasi untuk jenis, ketebalan dan gramasi benang pada kain.

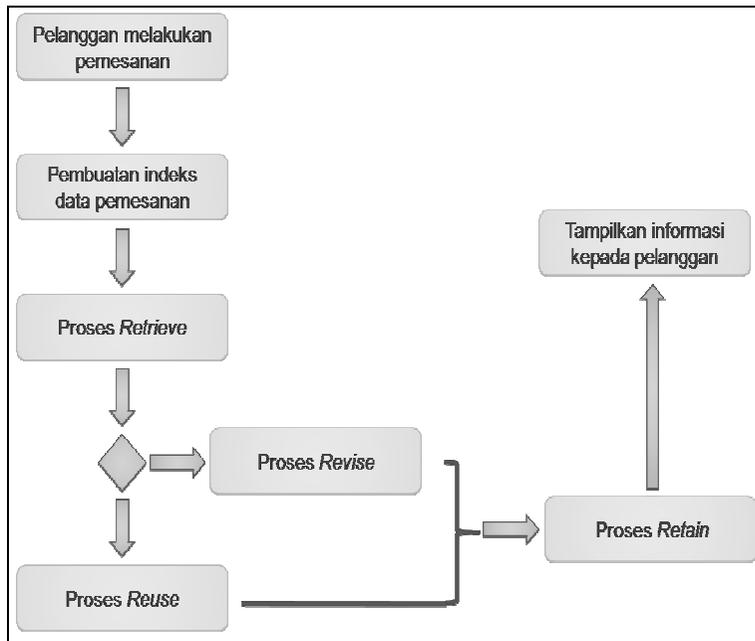
Pemodelan kasus direpresentasikan ke dalam basis data dengan *Entity Relationship Diagram* (ERD) selengkapnya dapat dilihat pada lampiran 1. Entitas pada aplikasi ini terdiri dari beberapa entitas, yaitu: pelanggan, pemesanan, pengiriman, kain, warna, finishing, kasus, *revise*, benang, mesin rajut, jadwal rajut, jadwal celup, *finishing*, mesin *finishing*, indeks dan referensi indeks. Dalam rancangan basis data tersebut, entitas pelanggan terhubung dengan entitas pemesanan. Entitas pemesanan terhubung kemudian dengan entitas pengiriman, kain, warna, *finishing*, kasus, dan *revise*. Entitas kain terhubung dengan entitas benang dan mesin rajut. Entitas mesin rajut terhubung dengan entitas jadwal rajut. Entitas warna terhubung dengan entitas mesin celup dan entitas mesin celup terhubung dengan entitas jadwal celup. Entitas *finishing* terhubung dengan entitas mesin *finishing* dan entitas mesin *finishing* terhubung dengan entitas jadwal *finishing*. Entitas kasus terhubung dengan entitas indeks. Selain itu terdapat entitas referensi indeks yang berisi data untuk membuat indeks.

Pada aplikasi ini, pada saat pelanggan melakukan pemesanan kain, sistem akan melakukan pembuatan indeks dari data pemesanan (lihat Kode Program pada Lampiran 3). Sistem kemudian menjalankan proses *retrieve*. Jika syarat *reuse* terpenuhi, sistem menjalankan proses *reuse*. Jika syarat tidak terpenuhi, sistem menjalankan proses *revise*. Dari proses *reuse* maupun *revise* akan dilanjutkan dengan proses *retain*. Selanjutnya sistem akan menampilkan informasi kepada pelanggan. Rancangan proses selengkapnya di dalam sistem dapat dilihat dalam Gambar 2.

Pada proses *revise*, *administrator* akan diberikan saran lama penyelesaian beserta dengan simpangannya. *Administrator* juga dapat melihat data 3 kasus dengan kedekatan tertinggi. Berdasarkan data – data tersebut, *administrator* dapat menentukan lama pemesanan kain.

Kedekatan Antar Kasus

Perhitungan untuk mendapatkan kesamaan antar kasus, sangat penting pada metode CBR terutama dalam proses *retrieval*. Efektifitas dari pengukuran kesamaan tergantung dari seberapa banyak kasus yang ditelaah, membantu dalam memecahkan masalah pada kasus yang baru. Ada dua pendekatan yang digunakan untuk melakukan perhitungan kesamaan yaitu pendekatan jarak dan struktur indeks. Pada aplikasi ini yang digunakan adalah pendekatan jarak dengan *Weighted Euclidean Distance*. [Pal2004]



Gambar 2 Proses CBR pada Aplikasi Pemesanan Kain

Pada pendekatan ini, pengukuran jarak dilakukan berdasarkan pada lokasi dari objek-objek pada *Euclidean Space*. Jarak dihitung sebagai akar dari jumlah selisih kuadrat antara dua buah objek. Untuk aplikasi ini, rumus yang digunakan adalah:

$$d_{pq}^{(w)} = \left[\sum_{j=1}^n w_j^2 (x_{pj} - x_{qj})^2 \right]^{1/2}$$

dengan,

d_{pq} = jarak antara kasus ke-p dan kasus ke-q.

\sum = jumlah dari parameter ke-1 sampai ke-n.

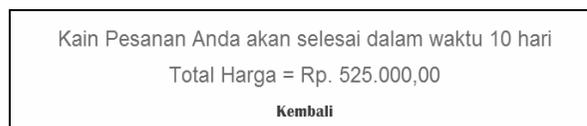
w = bobot (1,0.9,0.8,0.7,0.6,0.5,0.4,0.3).

Bobot disesuaikan dengan prioritas tertinggi parameter untuk pemesanan produk garmen, yaitu: jenis mesin, jenis benang, ketebalan, gramasi, kerapatan, *finishing*, warna, dan kuantitas.

x_{pj} = kasus ke-p parameter ke-j.

x_{qj} = kasus ke-q parameter ke-j.

j = banyaknya parameter (1,2,...,8)



Gambar 3 Tampilan Informasi Proses *Reuse*

Setelah jarak antara dua kasus didapat maka kedekatan antara dua kasus tersebut dihitung dengan menggunakan rumus :

$$SM_{pq}^{(w)} = \frac{1}{1 + \alpha d_{pq}^{(w)}}$$

dengan,

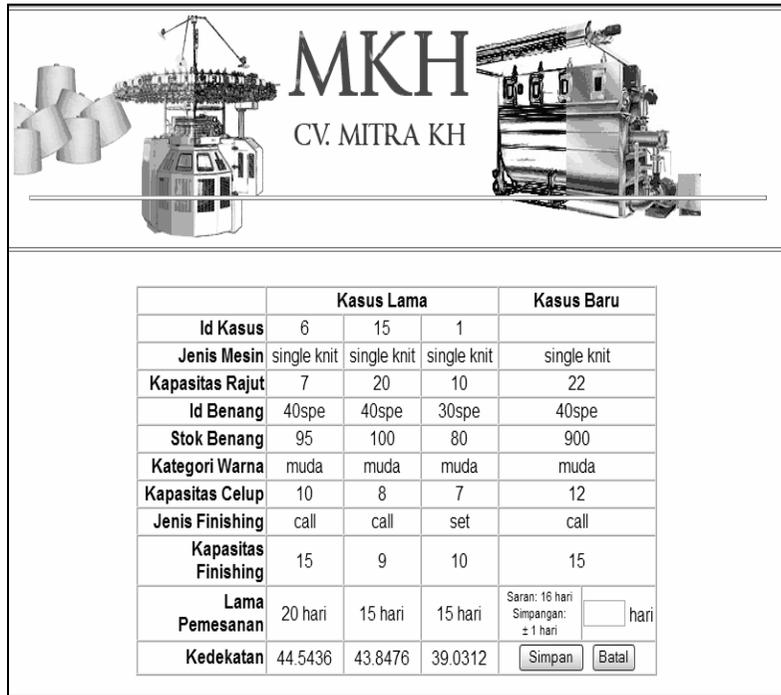
SM_{pq} = kedekatan kasus ke-p dan ke-q.

α = konstanta positif = 1 (jarak kasus dianggap tetap, karena memerlukan parameter yang sama)

d_{pq} = jarak antara kasus ke-p dan kasus ke-q.

Semakin tinggi nilai yang didapat maka kedekatannya pun akan semakin tinggi. Sebaliknya semakin rendah nilai yang didapat maka, kedekatannya pun semakin rendah. Pada aplikasi ini ditentukan sebuah batas, yaitu jika nilai yang didapat \geq nilai yang telah ditentukan maka kasus lama tersebut dapat langsung digunakan kembali untuk menyelesaikan kasus yang baru.

Pertama kali sistem akan mengambil data indeks kasus yang terdapat pada *database* untuk dicocokkan dengan indeks pemesanan. Sistem juga akan mengambil data bobot yang terdapat pada *database* untuk digunakan dalam proses perhitungan jarak antar kasus. Selanjutnya sistem akan melakukan pencocokan indeks pemesanan dengan indeks kasus yang terdapat pada *database*. Indeks pemesanan yang cocok dengan indeks kasus akan mendapat nilai 1, sedangkan untuk indeks yang tidak cocok akan mendapat nilai 0. Untuk parameter kapasitas rajut, stok benang, kapasitas celup, dan kapasitas *finishing* apabila memiliki indeks yang sama maka akan dicocokkan isi dari masing – masing parameter tersebut. Apabila isi parameternya sama maka akan mendapat nilai 1, apabila isi parameternya berbeda maka mendapat nilai 0.



Gambar 4 Tampilan Proses *Revise*

Setelah melakukan proses pencocokan sistem akan menghitung jarak antara pemesanan dengan kasus yang dicocokkan. Setelah jarak didapat, sistem akan menghitung kedekatan dari dua kasus tersebut. Kemudian tiga nilai kedekatan tertinggi yang didapat akan disimpan untuk digunakan pada proses *reuse* atau proses *revise*.

4. Metode *Program Evaluation and Review Technique (PERT)*

Program Evaluation and Review Technique merupakan metode untuk menghitung perkiraan waktu yang diperlukan dalam menyelesaikan suatu pekerjaan. Metode ini diterapkan untuk memberikan saran lama penyelesaian pesanan apabila sistem menjalankan proses *revise*. Saran tersebut diperlukan untuk membantu *administrator* dalam menentukan lama penyelesaian kain yang dipesan oleh pelanggan. Analisa waktu tersebut dihitung dengan menggunakan rumus :

$$T_E = \frac{(O + 4M + P)}{6}$$

dengan,

T_E = Perkiraan waktu.

O = Waktu minimum penyelesaian.

M = Perkiraan waktu paling baik untuk penyelesaian.

P = Waktu maksimum penyelesaian.

Pada metode PERT, terdapat simpangan dari waktu yang dibutuhkan untuk menyelesaikan suatu pekerjaan. Simpangan yang disebut Standar Deviasi dihitung dengan menggunakan rumus :

$$SD = \frac{P - O}{6}$$

dengan,

SD = Standar Deviasi

P = Waktu maksimum penyelesaian.

O = Waktu minimum penyelesaian.

Contoh penggunaan rumus pada metode PERT :

Terdapat 3 lama pemesanan yang berasal dari 3 kasus dengan nilai kedekatan tertinggi, yaitu :

Lama pemesanan kasus 1 : 27 hari.

Lama pemesanan kasus 2 : 16 hari.

Lama pemesanan kasus 3 : 23 hari.

$$T_E = \frac{(16 + 4(23) + 27)}{6}$$

$$T_E = 22.5 \approx 23 \text{ hari}$$

$$SD = \frac{27 - 16}{6}$$

$$SD = 1.83 \approx 2 \text{ hari}$$

5. Implementasi

Implementasi dilakukan dengan pendekatan berorientasi obyek, dengan keterkaitan antar kelas sebagai sebagaimana terlihat pada Lampiran 2. Contoh tampilan pemesanan dalam aplikasi web dapat dilihat pada Gambar 5 atau dalam aplikasi mobile GPRS pada Gambar 6.

Bahasa pemrograman yang digunakan pada pembuatan aplikasi pemesanan kain ini adalah PHP versi 5 dengan *database MySQL* versi 5.0.33. Perangkat lunak yang digunakan adalah *Adobe Dreamweaver CS3*, *Apache* versi 2.2.4, *phpMyAdmin* versi 2.9.2, *Internet Explorer* versi 7.0, *Openwave Phone Simulator* 7.0 versi 7.0.107.

Untuk melakukan pemesanan, pelanggan diminta untuk memasukkan data pemesanan yang terdiri dari jenis mesin, jenis benang, ketebalan, gramasi, kerapatan, *finishing*, warna, dan kuantitas. Apabila data belum lengkap maka pelanggan akan diminta untuk melengkapi data terlebih dahulu. Apabila data sudah lengkap maka akan dilakukan verifikasi data, jika data belum benar maka pelanggan dapat memperbaiki data terlebih dahulu, jika data sudah benar maka data pemesanan akan disimpan ke tabel pemesanan. Selanjutnya sistem akan membuat indeks data pemesanan untuk kemudian dicocokkan dengan indeks pada tabel kasus dan dihitung kedekatannya.

Data dengan kedekatan tertinggi akan diambil dan dicek nilai kedekatannya. Apabila nilai lebih kecil dari 50% maka data perlu di-*revise*, jika nilai lebih besar sama dengan 50% maka prioritas akan dicek apakah memenuhi syarat atau tidak. Apabila tidak memenuhi syarat maka data perlu di-*revise* dan apabila data sudah memenuhi syarat maka data lama pemesanan akan diinformasikan kepada pelanggan.

6. Kesimpulan dan Saran

Hasil yang dicapai dari penerapan *Case Based Reasoning* sesuai dengan yang diharapkan, yaitu sistem dapat melakukan proses *retrieve* yaitu proses pengambilan data – data yang memiliki kesamaan dengan permasalahan yang sedang dihadapi. Selanjutnya data tersebut akan dicocokkan dan dihitung kedekatannya sehingga dapat diketahui data yang memiliki kesamaan tertinggi. Data dengan kesamaan tertinggi tersebut akan diambil solusinya. Selanjutnya sistem akan menjalankan proses *reuse*.

Proses *reuse* adalah proses dimana setelah menemukan solusi, sistem akan melakukan pengecekan apakah solusi yang telah didapat tersebut akan diterapkan kembali atau tidak. Apabila solusi tersebut telah memenuhi persyaratan maka solusi yang telah didapat tersebut akan digunakan untuk memecahkan masalah yang sedang dihadapi. Akan tetapi apabila solusi tersebut belum memenuhi persyaratan maka sistem akan menjalankan proses selanjutnya yaitu proses *revise*.

Proses *revise* adalah proses yang dijalankan untuk memperbaiki solusi yang telah didapat dari proses *reuse*. Perbaikan solusi dilakukan dengan mengacu kepada data pemesanan lama yang memiliki kedekatan tertinggi dengan bantuan saran lama waktu pemesanan yang dihitung menggunakan metode PERT. Selanjutnya proses *retain* akan dijalankan. Proses *retain* adalah proses penyimpanan data yang dilakukan apabila data tersebut belum terdapat pada *database*. Apabila data sudah terdapat dalam *database*, maka data tidak akan disimpan kembali agar tidak terjadi duplikasi data pada *database*.

Saran pengembangan untuk mendapatkan hasil yang lebih akurat, maka parameter – parameter pencocokan yang digunakan perlu dilengkapi.



Gambar 5 Form Pemesanan

Parameter - parameter tersebut memberikan pengaruh terhadap tingkat akurasi dari informasi lama pemesanan yang dihasilkan. Semakin tinggi kecocokan antara data pada tabel kasus dengan data pemesanan, maka semakin tinggi juga kedekatan antara kedua kasus yang berarti data pemesanan dengan data kasus memiliki keadaan yang hampir sama dan informasi yang dihasilkan akan semakin baik. Parameter – parameter yang dapat ditambahkan untuk pemesanan kain antara lain adalah: get, gramasi, dan kerapatan.



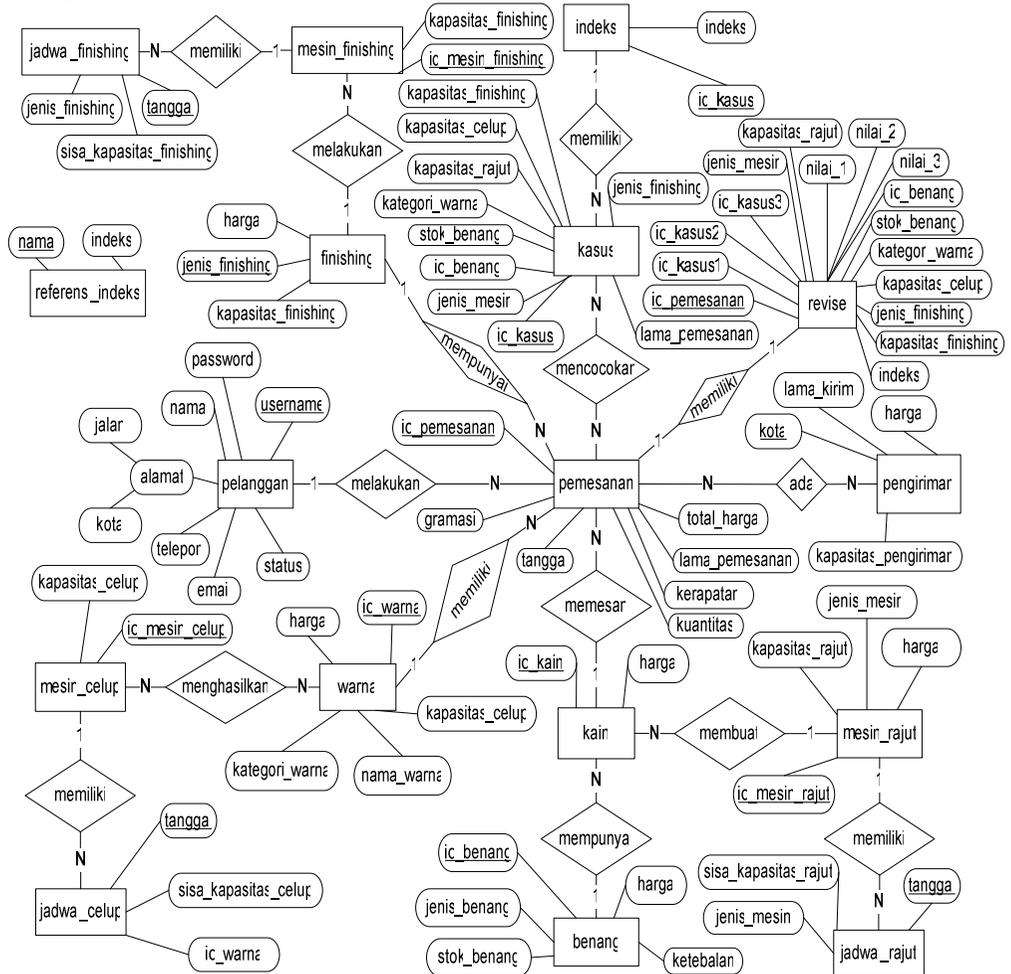
Gambar 6 Form Pemesanan Mobile

Daftar Pustaka

- [Aam94] Aamodt, Agnar, Enric Plaza. 1994. *Case Based Reasoning : Foundational Issues, Methodological Variations, and System Approaches*. Available : www.iiia.csic.es/People/enric/AICom.html. Accessed : 9 Agustus 2007
- [Ash88] Ashley, K.D. (1988). Arguing by Analogy in Law: A Case-Based Model. In D.H. Helman (Ed.), *Analogical Reasoning: Perspectives of Artificial Intelligence, Cognitive Science, and Philosophy*. D. Reidel.
- [Cas07] Castro, Elizabeth. 2007. *HTML,XHTML & CSS*. Peachpit Press.
- [Cog05] Coggeshall, John. 2005. *PHP 5 Unleashed*. Sams.

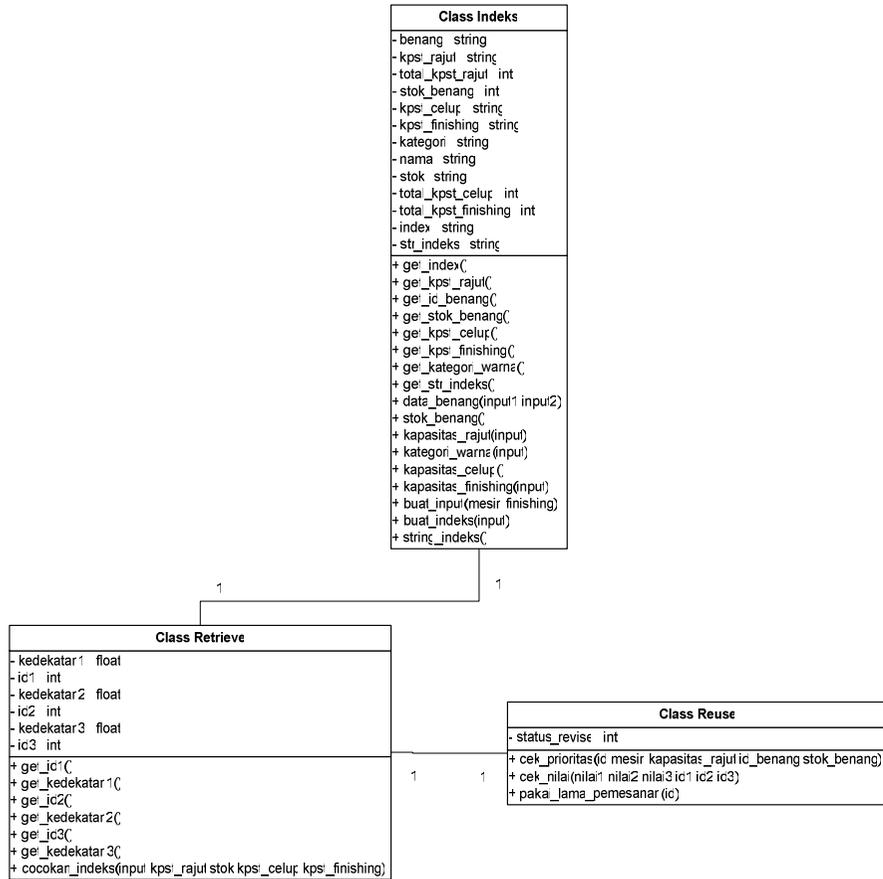
- [Hoj07] Hojtsy, Gabor. 2007. *PHP Manual*. Available : <http://www.php.net/docs.php>. Accessed : 9 Agustus 2007.
- [Kol83] Kolodner, J. L. (1983a). Maintaining Organization in a Dynamic Long-Term Memory. *Cognitive Science*, 7(iv): pp.243-80.
- [Nar05] Naramor, Elizabeth, dkk. 2005. *Beginning PHP 5, Apache, and MySQL Web Development*. Wiley Publishing.
- [Pal04] Pal, Sankar K., Simon C. K. Shiu. 2004. *Foundation of Soft Case-Based Reasoning*. John Wiley & Sons.
- [Per08] _____. (2008) *PERT/CPM for Project Scheduling & Management* Available : <http://www.interventions.org/pertcpm.html>. Accessed : 30 Januari 2008.
- [Rus03] Russel, Stuart J., Peter Norvig. 2003. *Artificial Intelligence a Modern Approach*. Prentice Hall.
- [Wat94] Ian Watson & Farhi Marir. 1994. Case-Based Reasoning: A Review. *Knowledge Engineering Review*, Vol 9 No. 4.

Lampiran 1: Diagram Entitas Relasional



Lampiran 2: Keterkaitan Antar Kelas

Pada aplikasi ini terdapat tiga *class* utama, yaitu *class* indeks, *class* retrieve, *class* reuse. *Class* retrieve akan mengambil data indeks, kapasitas rajut, stok benang, kapasitas celup, dan kapasitas finishing dari *class* indeks. Kardinalitas antara *class* retrieve dan *class* indeks adalah 1 ke 1. *Class* reuse akan mengambil data id 1, id 2, id 3, kedekatan 1, kedekatan 2, dan kedekatan 3 dari *class* retrieve. Kardinalitas antara *class* reuse dan *class* retrieve adalah 1 ke 1.



Lampiran 3: Cuplikan Program Pencocokan Indeksasi Kasus

```
Function cocokan_indeks($input, $kpst_rajut, $stok, $kpst_celup, $kpst_finishing)
{
    $indeks_pesan = $input;
    ► ambil data indeks dari database
    $ambil_indeks_db = mysql_query("select * from tbl_indeks");
    while($hasil_indeks_db= mysql_fetch_assoc($ambil_indeks_db))
    {
        $id_kasus[] = (int)$hasil_indeks_db["id_kasus"];
        $indeks_db[] = $hasil_indeks_db["indeks"];
    }
    ► ambil bobot dari database
    $ambil_bobot = mysql_query("select * from tbl_pengaturan");
    $hasil_bobot = mysql_fetch_row($ambil_bobot);
    $count_id_kasus = count($id_kasus);
    ► cocokan indeks pemesanan dengan indeks kasus pada database
    for($i = 0; $i < $count_id_kasus; $i++)
    {
        $indeks_kasus = str_split($indeks_db[$i],1);
        $count_indeks_kasus = count($indeks_kasus);
        ► beri nilai untuk indeks yang sama
        for($j = 0; $j < $count_indeks_kasus; $j++)
        {
            if($indeks_pesan[$j] == $indeks_kasus[$j])
                $nilai[] = 1;
            else
                $nilai[] = 0;
        }
        $count_nilai = count($nilai);
        ► cocokan isi parameter(kapasitas rajut, stok benang,  
kapasitas celup, kapasitas finishing) dan hitung jarak
        for($k = 0; $k < $count_nilai; $k++)
        {
            if($k % 2 == 1 && $nilai[$k] == 1)
            {
                $ambil_nilai_db = mysql_query("select * from tbl_kasus where
                id_kasus='$id_kasus[$i]'");
                $hasil_nilai_db = mysql_fetch_row($ambil_nilai_db);
                $x = $k + 1;
                $nilai_db = $hasil_nilai_db[$x];
                if($k == 1 && $kpst_rajut == $nilai_db)
                    $jarak += pow((float)$hasil_bobot[2],2);
                if($k == 3 && $stok == $nilai_db)
            }
        }
    }
}
```

```
$jarak += pow((float)$hasil_bobot[4],2);
if($k == 5 && $kpst_celup == $nilai_db)
    $jarak += pow((float)$hasil_bobot[6],2);
if($k == 7 && $kpst_finishing == $nilai_db)
    $jarak += pow((float)$hasil_bobot[8],2);
}
else
{
    if($nilai[$k] == 1)
    {
        $a = $k + 1;
        $jarak += pow((float)$hasil_bobot[$a],2);
    }
}
}
}
► hitung kedekatan
if($jarak >= 3.8)
    $kedekatan_temp = 100;
else
    $kedekatan_temp = ((1/(1+(1*(sqrt(3.8-$jarak)))))) * 100);
$jarak =0;
}
}
```