



Software Structure

Erick Kurniawan, M.Kom



Topik Bahasan

- User Database
- Content Database
- User / Content Map
- User / User Map



Pengantar

- User database dapat diimplementasikan sebagai kumpulan dari SQL table didalam relational database management system
- Table untuk user tidak boleh terpisah dengan table yang lain (yang diimplementasikan untuk modul yang berbeda)
- Table table tersebut juga harus dimiliki oleh user yang sama dan dalam satu tablespace (oracle)
- Disisi lain user database bisa dibuat external (diluar source dari OLC)
- Dimana employees dapat melakukan pengecekan autentikasi dengan LDAP server



Pengantar

- Contoh yang lebih modern, dimana database dapat dipisahkan di Internet menggunakan Web Services
- Microsoft Hailstorm, menawarkan layanan user database
- Data siswa dari sekolah yang berbeda mungkin mempunyai struktur table yang sama
- Biarpun tersimpan pada komputer yang berbeda (secara fisik) tapi dapat diakses menggunakan Web Services



User Database

- Paling tidak user database mencatat real name dan email address dari user
- Semakin banyak user yang dicatat, di-autentikasi, dan dihitung maka semakin berhasil untuk mengembangkan OLC
- Menghindari situasi yang tidak diinginkan
- Situasi dimana anonymous user saling memberi komentar tanpa tahu nama sebenarnya (seperti di www2.ukdw.ac.id)
- Menyulitkan pekerjaan administrator (harus mendelete / mengupdate pesan)



User Database

- User database seharusnya juga memiliki kemampuan untuk menyimpan personal URL untuk tiap usernya (Yahoo Geocities)
- Mengatasi masalah URL rot, user database membutuhkan extra field untuk meyimpan track
- Ketika robot mengunjungi URL tersebut dan tidak dapat mengaksesnya lebih dari 1 minggu, maka dapat diasumsikan bahwa URL tersebut of out date dan tidak mencantumkan-nya dalam list
- Informasi yang bersifat privacy pada database user seharusnya tidak ditampilkan (mis: alamat email)



Content Database

- Isi / Content dari OLC diantaranya adalah pertanyaan dan jawaban dari forum diskusi
- Dimulai dari membangun tabel untuk discussion forum posting
- OLC menawarkan published article yang penyimpanan datanya berbeda dengan user-contribute question
- Magnet Content (artikel yang menarik)
- Programmer juga harus membuat beberapa table untuk menyimpan data article



Content Database

- Situs OLC yang baik juga menawarkan kepada user untuk memberi kontribusi berupa komentar pada article (disimpan pada tabel yang terpisah)
- Beberapa services yang disediakan dan datanya disimpan pada content database
 - Versioning dari content
 - reply, comment, attachment
 - Content yang disetujui atau ditolak oleh site moderator
 - Untuk siapa content tersebut ditampilkan (public, group, atau user tertentu)
 - User siapa yang berhak untuk mengedit content



Content Database

- Siapa yang hanya berhak melihat, mengedit content
- Siapa yang berhak memberi comment, siapa yang harus mereview comment
- Waktu dari content yang ditampilkan, berapa lama? Kapan expired?
- Description, keyword dari content
- Siapa yang menulis? , group, user?
- Rate-nya (sering diakses atau tidak)



User Content Map

- OCL juga harus dapat melakukan pencatatan log dari apa yang dilakukan oleh user:
 - User #21 menulis comment #37 pada artikel #529
 - User #192 ask question #512
 - User #451 posted Answer #3 untuk question #924
 - User #1392 sudah membaca artikel #456
 - User #8923 minta di beritahu dengan alert jika article #223 berubah
- Kita perlu menyimpan record authorship, mis: untuk menawarkan service email notification ketika seseorang meresponse pertanyaan pada article, diperlukan untuk menyimpan authorship



User / Content Map

- Apakah record tentang user yang mengakses content / readership perlu disimpan? Perlu jika anda ingin robot mengirimkan pesan secara otomatis (mis: prof bisa tahu mhs yang tidak pernah membaca materi)
- Dengan me-record readership kita bisa tahu kebiasaan, ketertarikan dari user (Knowledge Management System)
- Bisa memantau aktifitas user (mis: user tidak pernah aktif, baca artikel, dll)
- Mempertimbangkan KM, awalnya database hanya kosong dan user masih sedikit, scanning



User / Content Map

- Mempertimbangkan KM, awalnya database hanya kosong dan user masih sedikit
- Ketika kita mencari judul dari semua contributed content hanya membutuhkan sedikit waktu
- Bagaimana setelah 5 tahun?
- Database berisi 100.000 document dan 10.000 active user yang menambahkan beberapa ratus document setiap hari
- Jika user (Jane) ingin mencari content yang baru, mungkin membutuhkan waktu 30 menit



User / Content Map

- Bayangkan jika kita menyimpan data-data content kesukaan jane.
- Ketika jane searching maka ditampilkan data-data yang disukai jane saja
- Disimpan dengan metode scoring, mencatat score content apa yang paling banyak diakses kemudian membandingkannya



User / User Map

- Relationship diantara user juga penting ketika komunitas semakin berkembang
- Grouping user adalah operasi fundamental yang dilakukan dalam User database
- Dapat melakukan group per departemen