

Data Driven

Djoni Darmawikarta

djoni_darmawikarta@yahoo.ca

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Dalam tulisan ini, teknik data driven dijelaskan dengan menggunakan contoh yang relatif sederhana. Setelah memahami konsepnya melalui contoh ini, teknik data driven ini bisa diterapkan untuk kasus-kasus yang kompleks.

Dalam contoh ini, kita harus membuat program untuk menghitung bonus karyawan. Besar bonus masing-masing karyawan tergantung gaji pokok dan jenis tanggung-jawabnya.

Data profil_karyawan

nama_karyawan	gaji_pokok	tanggung_jawab
Andi Halim	1000	Sales manager
Budi Sadikin	750	Senior salesman
Chandra Kusuma	500	Junior salesman
Dedi Wirobroto	550	Junior salesman
Eko Santoso	560	Junior salesman
Farid Abulkarim	560	Junior salesman
...		...

Teknik yang sering kita pakai adalah melaksanakan seluruh logic perhitungan didalam program (hardcoding) Misalnya dengan menggunakan if-then-else sebagai berikut.

Ambil tanggung_jawab dari profil_karyawan

If tanggung_jawab = "Sales manager"

Then bonus = gaji_pokok * 0.10

Else

If tanggung_jawab = "Senior salesman"

Then bonus = gaji_pokok * 0.08

Else ... dan seterusnya ...

Apa yang terjadi bila ada perubahan pada faktor ketergantungan yang merupakan parameter-parameter perhitungan tersebut? Misalnya, nama jenis tanggung_jawab

dan/atau faktor pengali berubah, atau ada jenis tanggung jawab baru yang juga perlu dihitung bonusnya?

Kita harus selalu menyesuaikan program dengan perubah-perubahan ini!

Data driven

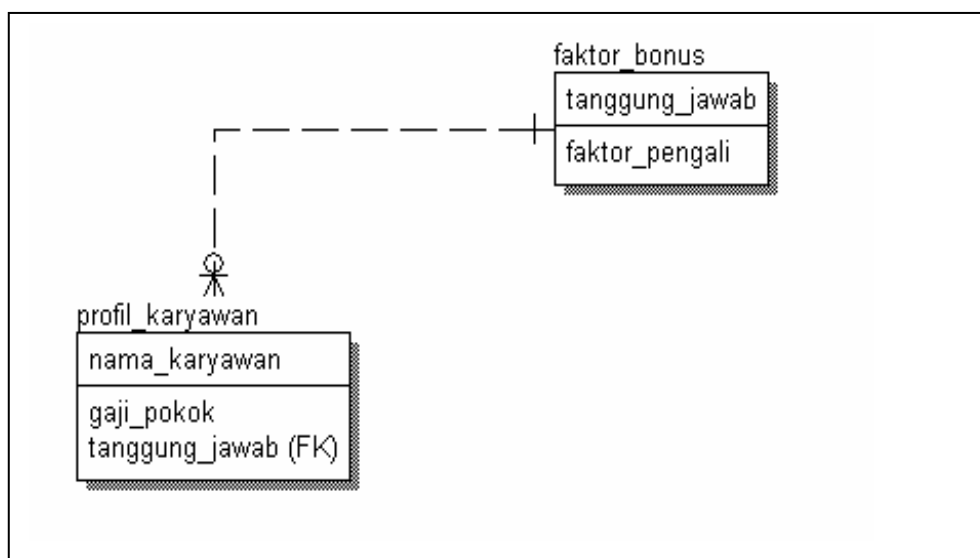
Data driven merupakan solusi efektif untuk menghindari masalah-masalah hardcoding seperti dicontohkan diatas.

Kita membuat tabel faktor_bonus sebagai berikut.

tanggung_jawab	faktor_pengali
Sales manager	0.10
Senior salesman	0.08
Junior salesman	0.06
...	...

Kita implementasikan dalam database relasional dan hubungkan kolom tanggung_jawabnya dengan kolom yang sama di tabel profil_karyawan (di-migrate-kan sebagai foreign key) Dengan demikian data tanggung_jawab dikedua tabel selalu sinkron (integritas terjamin)

Berikut diagram ER (entity relationship) kedua tabel ini (menggunakan notasi IE, Information Engineering)



Maka struktur program kita, yang menggunakan fungsi SQL join, menjadi generik dan robust (tahan perubahan) sebagai berikut.

```
Ambil tanggung_jawab dari profil_karyawan -- digunakan pada WHERE clause
berikut --
SELECT gaji_pokok, faktor_pengali FROM profil_karyawan, faktor_bonus
      WHERE profil_karyawan.tanggung_jawab = faktor_bonus.tanggung_jawab
Bonus = gaji_pokok * faktor_pengali
```

Kita bisa membuatkan program sederhana untuk mereka (user), yang memang bertanggung-jawab tentang bonus, agar mereka sendiri bisa memelihara data faktor bonus sesuai kebutuhannya.

Dengan teknik data driven kini program kita jadi stabil! Dan, para programmer berkesempatan dengan tenang mengerjakan tugas teknis lain daripada senantiasa harus merubah program yang sama sepanjang masa.