

BAB 15 PROTOTIPE

Bekerja dengan Model Pertama

15.1. PENDAHULUAN

Siapa pun yang pernah menyelesaikan proyek software akan sependapat, bahwa masalah pertama adalah memperoleh kebutuhan dari user. Permasalahan kedua adalah berdasarkan persetujuan Spesifikasi Fungsional (FS). FS mencoba untuk menggambarkan sistem yang berbasis grafik dan narasi, tetapi gambar dan penjelasan tidak dapat menerangkan cara sistem tersebut berjalan, berlaku, dan mempengaruhi bisnis user. Sebagai tambahan, FS biasanya menimbulkan kesalah pahaman (jika dibaca semuanya).

Kesalah pahaman antara user dan analis mengakibatkan perubahan yang berarti atau sistem tidak akan pernah sempurna dalam pelaksanaannya atau sekaligus ditolak. Prototipe dapat memecahkan masalah ini untuk tipe-tipe tertentu dalam sistem.

15.2. TEORI DI BELAKANG PROTOTIPE (THE THEORY BEHIND PROTOTYPING)

Apakah anda membeli mobil dari brosur penjualan ?

Seperti halnya anda tidak dapat menilai sebuah mobil tanpa mencobanya, user juga tidak dapat menilai dari Spesifikasi Fungsional, bagaimana sistem akan berlaku dan berjalan. Tetapi jika user dapat melihat, menyentuh dan menggunakan 'mode' atau prototipe dari tujuan sistem, ia dapat langsung menilai kegunaan sistem. Jika perubahan diperlukan prototipe dapat dimodifikasi, memungkinkan dimodifikasi beberapa kali sampai keadaan yang ditetapkan user.

Keuntungan dari prototipe

- Menghasilkan syarat yang lebih baik dari produksi yang dihasilkan oleh metode 'spesifikasi tulisan'.
- User dapat mempertimbangkan sedikit perubahan selama masih bentuk prototipe.
- Memberikan hasil yang lebih akurat dari pada perkiraan sebelumnya, karena fungsi yang diinginkan dan kerumitannya sudah dapat diketahui dengan baik.
- User merasa puas. **Pertama**, user dapat mengenal melalui komputer. Dengan melakukan prototipe (dengan analisis yang sudah ada), user belajar mengenai komputer dan aplikasi yang akan dibuatkan untuknya. **Kedua**, user terlibat langsung dari awal dan memotivasi semangat untuk mendukung analisis selama proyek berlangsung.

15.3. METODE PROTOTIPE (THE PROTOTYPING METHOD)

Langkah-langkah pembuatan prototipe :

Langkah Pertama

Permintaan bermula dari kebutuhan user.

Langkah Kedua

Bangunlah sistem prototipe untuk menemukan kebutuhan awal yang diminta.

Langkah Ketiga

Biarkan user menggunakan prototipe. Analis harus memberikan pelatihan, membantu dan duduk bersama-sama dengan user, khususnya untuk pertama kali. Anjurkan perubahan. User harus melihat fungsi-fungsi dan sifat dari prototipe, lihat bagaimana ia memecahkan masalah bisnis dan mengusulkan perbaikan.

Langkah Keempat

Implementasikan saran-saran perubahan.

Langkah Kelima

Ulangi langkah ketiga sampai user merasa puas.

Langkah Keenam

Merancang dan membangun suatu sistem akhir seperti sebelumnya.

**15.4. SISTEM YANG BERMANFAAT DARI PROTOTIPE
(SYSTEMS THAT BENEFIT FROM PROTOTYPING)**

Sejak kebutuhan (baca Spesifikasi Fungsi) pada umumnya berhubungan dengan pandangan user terhadap sistem, hanya dengan prototipe tampilan bagi user sudah cukup untuk memeriksa yang dibutuhkan. Menu-menu, bentuk tampilan input, tampilan keluaran, atau laporan yang dicetak, pertanyaan-pertanyaan, pesan-pesan merupakan calon yang ideal untuk prototipe.

Di lain pihak, perhitungan yang rumit, kumpulan *update* data dan *realtime*, dan sistem yang bersifat scientific sangat sulit untuk dijadikan model.

Sistem yang paling sesuai untuk prototipe adalah satu dari banyak hal yang bergantung pada sistem input/output dari user. Sistem dengan transaksi on-line dikendalikan melalui menu, layar, formulir, laporan, daftar dan perintah.

15.5. SOFTWARE UNTUK PROTOTIPE (SOFTWARE FOR PROTOTYPING)

Apa yang harus disediakan dari paket software prototipe ?

Produk yang baik harus menyediakan 7 bagian, yaitu :

1. Pembuatan menu yang cepat dan mudah.

Pada menu harus terdapat sub menu, formulir, laporan, program protipe, dan menyediakan bantuan secara on-line untuk pemilihan menu dan prompt.

Lihat Gambar 15.1. Konstruksi Program Menu

2. Pembuatan tampilan input dan output.

Anda harus dapat mewarnai bentuk tampilan dengan menempatkan kursor pada lokasi yang dituju (mouse merupakan alat yang terbaik untuk melakukannya), ketik nama field, spesifikasikan ketentuan untuk mengedit berdasarkan panjang field, menyertakan alfanumerik, jarak angka-angka yang diperbolehkan, kesalahan dan pesan-pesan bantuan, dll.

Lihat Gambar 15.2. Cara kerja field program aplikasi

3. Penyelarasan.

Anda dapat menjelaskan bentuk sebuah laporan yang dicetak dengan mudah. Bagian-bagian untuk merinci pembuatan laporan adalah judul, catatan kaki, field mana yang diletakkan (yang paling baik adalah jika program menampilkan semua field yang dikenal), header kolom, pengelompokkan, pengurutan, serta sub dan grand total. Pada umumnya seseorang harus dapat melaporkan item yang dipilih saja.

Lihat Gambar 15.3. Penggunaan sebuah program pembuatan laporan

4. Software harus menghasilkan sebuah Data Dictionary secara otomatis.

Data Dictionary (DD) menyimpan informasi seperti layar, laporan atau formulir, tetapi yang paling penting DD menyimpan setiap informasi pada setiap field, termasuk panjang field, pengeditan dalam setiap laporan, dan format field yang digunakan.

DD adalah inti dari setiap produk dan sudah seharusnya setiap alat prototipe menggunakan DD untuk mengecek apakah fieldnya digunakan secara konsisten pada setiap tampilan, dan apakah dapat menyimpan pengetikan berulang jika field muncul lebih dari 1 kali.

5. Software harus dapat menyusun database sesuai harapan.

Ketentuan tampilan input seperti yang digunakan pada gambar 15.2 menjelaskan tentang peralatan daftar format. Software harus menyusun database dan selanjutnya mengizinkan user memasukkan data menggunakan formulir input. Produk-produk yang baik mengizinkan user untuk mengoptimalkan database dengan menentukan format dan kunci recordnya.

6. Mencari hasil dengan query on-line secara tepat ke data yang didaftar pada database.

Anda harus mampu melakukan pencarian dengan mudah, penyusunan, pemilihan dan menampilkan record.

7. Apakah yang dibutuhkan meliputi logika yang rumit atau perhitungan yang diperlukan prototipe ?

Walaupun tidak penting, program yang baik mempunyai bentuk struktur sederhana bahasa pemrograman untuk mengizinkan anda melakukan pemrosesan khusus, waktu kejadian, prosedur-prosedur otomatis, dsb.

Prototipe sebagai bagian dari CASE

Prototipe adalah metode untuk mengotomatisasi fase definisi dan analisis, jadi merupakan bagian dari CASE (*Computer Aided Software Engineering*). Tetapi prototipe memberikan masukan pada tingkat pengotomatisasian selanjutnya dengan baik.

Jika langkah selanjutnya adalah membangun sistem nyata dalam bahasa pemrograman generasi ketiga, produk prototipe harus dapat menyediakan pada anda sarana untuk mencetak semua item yang telah diketahui, semua menu, semua bentuk laporan dan perintah-perintah.

Produk yang terbaik adalah yang mengijinkan user untuk mencetak dokumen yang tersusun secara logika dengan bab dan bagian yang dapat digunakan seperti dokumen Spesifikasi Fungsional (FS). Bahkan beberapa diantaranya menyediakan pengolah kata untuk memasukkan teks keterangan di antara item-item. Produk tersebut harus pula dapat mencetak kamus data, yang dapat menghemat banyak jam kerja.

Jika disain dan programnya menggunakan bahasa pemrograman generasi ke-empat atau peralatan CASE yang terintegrasi (lihat Bab 6), anda harus dapat memasukkan hasil dari prototipe secara langsung ke dalam peralatannya. Peralatan CASE yang paling baik akan secara otomatis menyusun database dan bahkan kode untuk aplikasi akhir.

Seberapa cepat anda melakukan prototipe ?

Software untuk melakukan prototipe harus dapat membuat anda mampu menyusun model permulaan dari suatu sistem secara cepat. Khususnya, anda dapat membangun prototipe pertama dari bentuk terkecil hingga bentuk yang sedang hanya dalam waktu 2 s/d 3 minggu !. Anda juga harus dapat mengimplementasikan perubahan dengan cepat.

Ini akan memakan waktu beberapa menit untuk membuat perubahan, seperti memindahkan sebuah field pada sebuah form. Ini akan

memakan waktu kurang dari 1 jam untuk menentukan sebuah menu atau form baru, dan hampir beberapa jam untuk membuat sebuah file baru atau menyusun ulang sebuah file yang sudah ada untuk sebuah field, key atau akses baru.

Untuk mencapai kecepatan ini perlu mempunyai sebuah *user interface* yang sederhana dan logis. Sebenarnya jika pembuatan prototipe software cukup sederhana, para pengembang dapat melatih user untuk menggunakannya. User kemudian akan membangun prototipe dan menghubungi pengembang ketika siap untuk berubah menjadi sistem akhir.

15.6. DIMANAKAH PROTOTIPE DITEMPATKAN DALAM 7 FASE ?

Sekarang anda mungkin merasakan bahwa metode pembuatan prototipe dapat menggantikan dengan sempurna 7 fase mendekati pembangunan proyek (dan mungkin juga merasa diganggu bahwa saya membuat anda membaca bagian I buku ini sebelum menceritakan pada anda tentang pembuatan prototipe !) Jangan takut pembuatan prototipe hanya menggantikan beberapa bagian dari Fase Definisi dan Analisis.

Lihat Gambar 15.4. Urutan kejadian di dalam 7 fase antara metode lama dengan metode baru (prototipe)

Fase definisi dan analisis akan memakan waktu yang lebih sedikit dari cara baru, karena sebuah *deadline* akan ditentukan waktunya.

15.7. BEBERAPA PRODUK YANG HARUS DILIHAT (SOME PRODUCTS TO LOOK AT)

Excelerator sebagai alat prototipe

Dalam bab 6 kita telah melihat sebuah alat analisis yang baik disebut Excelerator (referensi 2.2), tetapi alat ini mempunyai bentuk yang membuat alat prototipe yang baik. Meliputi menu, form, dan fasilitas untuk membuat laporan. Yang terpenting memelihara sebuah kamus data, memungkinkan anda untuk mencetak sebuah organisasi logik yang berfungsi sebagai dokumen yang khusus, yang terdiri dari semua jenis yang didefinisikan dengan proses kata yang dimasukkan dalam paragraf untuk ukuran yang baik.

Bahasa Generasi Ke-empat sebagai peralatan prototipe

Suatu saat kita memerlukan layar yang rumit, perhitungan khusus, prosedur otomatis, dan bentuk laporan yang unik, yang kebanyakan prototipe tidak mempunyai kemampuan untuk melakukan semua itu. Ada sebuah jenis produk yang disebut *Fourth Generation Language* (4GL) yang mempunyai kemampuan untuk melakukan hal tersebut. Kita akan membicarakan hal ini di bab lain sebagai alat untuk mengembangkan semua sistem, tetapi hampir semua 4GL dapat digunakan sebagai peralatan prototipe yang baik.

15.8. KESIMPULAN (CONCLUSIONS)

Prototipe telah digunakan dengan sukses untuk diimplementasikan ke beberapa sistem software, yang meliputi user interface melalui menu, layar, laporan, dan transaksi on-line. Sistem yang berorientasi bisnis sangat cocok untuk metode ini. Sistem *real-time* dan *scientific* akan sedikit berkurang ketika menggunakan metode ini.

