

BAB 15

BENTUK DASAR

Bekerja dengan modal yang pertama

15.1 **PERKENALAN**

Orang-orang terkenal yang direkrut

Saya sering mendengar dan mengikuti percakapan didalam analisis. Analisis sangat potensial bagi pengguna. Siapa yang tidak menggunakan manfaat dengan tidak menggunakan eksperimen komputer. Analisis bertanya “mengapa kamu tidak mendaftarkan sebagai siswa, apakah kamu suka jaringan dari PC kamu ke Departemen kecil dan melihat program windows di perusahaan kredit atau keduanya?”. Pengguna menjawab dengan cara yang membingungkan, “ saya selalu mencari windows dikantor saya !”.

Siapun yang pernah melakukan proyek software akan sependapat bila masalah pertama didapat dari pemberi syarat dan pengguna masalah kedua meminta pendapat bagaimana membangun Fungsional Specification (FS). Percobaan FS diperkenalkan dengan sistem yang menggunakan grafik dan cerita. Tapi gambar dan penjelasan tidak dapat menerangkan jalan dengan cara merasakan, melakukan, berkelakuan, dan mempengaruhi didalam pengguna bisnis. Sebagai tambahan FS selalu salah paham (jika dibaca semua)

Kesalahpahaman diantara pemakai dan hasil analisis yang sangat mahal atau sistem yang tidak pernah lengkap dilaksanakan atau satu yang ditolak. Bentuk keaslian dapat memecahkan masalah-masalah untuk type sistem yang pasti.

15.2 TEORI DI BELAKANG BENTUK DASAR

Dapatkan kamu membeli mobil baru dari penjualan brosur

Kamu tidak bisa menjadi hakim dari mobil tersebut tanpa melalui ujian mengendarai, Pemakai tidak dapat menjadi hakim dari FS bagaimana sistem akan merasakan dan berkelakuan, tapi bila pemakai dapat melihat, menyentuh dan menggunakan model atau bentuk dasar yang diusulkan bentuk sistem., ia dapat siap menggunakan sistem yang berguna. Jika keperluan dari bentuk dasar diganti dengan caa dimodifikasi, barangkali beberapa waktu sampai pemakai kebagian “ya, ini adalah tipe dari sistem yang dapat memecahkan masalah saya!” kemudian menghasilkan hasil dan model yang baik sekali disistem akhir.

Keuntungan dari bentuk dasar

Hasil sistem bentuk dasar ini lebih baik syaratnya daripada produk yang ditulis dengan metode spesifikasi. Sedikit perubahan pasti terjadi. Kita berharap, pemakai hanya sedikit merubah selama itu masih bentuk dasar. Metode bentuk dasar akan berhasil dan lebih akurat didalam taksiran daripada sebelumnya, karena fungsi dan syarat dari kekomplekan akan lebih baik.

Manfaat lain yang sangat menggembirakan pemakai pertama kali, ia dapat dikenalkan secara jinak ke komputer, dengan perencanaan bentuk dasar pertama (dengan analisis memegang tangan), ia belajar tentang komputer dan aplikasi yang dibuat olehnya. Kedua, ia akan terlibat baik dari mulai dan motivasi memberi semangat ke analisis untuk durasi suatu proyek.

15.3 METODE BENTUK DASAR

Langkah-langkah bentuk dasar

Enam langkah yang harus diikuti

Langkah pertama : Ikuti syarat inisial dari pemakai dengan metode lama. Bentuk definisi kamu akan berkata kepemakainya, “kamu harus memberi syarat untuk 5 tahun dari tanggal X, jika tidak kami maju

kedepan dengan anggapan kamu perlu dan kamu akan dapat merubah apa saja (dengan mudah)” dengan metode (bentuk dasar) baru kamu akan berkata, "berilah kami apa yang dipikirkan kamu yang diperlukan. Dan kamu akan merubah pikiranmu selama kamu bisa (hampir)”.

- Langkah kedua : Bangunlah sistem bentuk dasar untuk menginginkan syarat inisial.
- Langkah ketiga : Ikuti pemakai dengan bentuk dasar. Analisis harus diajarkan, bantuan dan duduk dengan pemakai, terutama waktu pertama mendorong untuk mengganti. Pemakai harus mengganti fungsi-fungsi dan kelakuan dari bentuk dasar. Melihat bagaimana masalah bisnis dipecahkan dan memberi kesan memperbaiki.
- Langkah keempat : Melaksanakan dengan memberi kesan mengganti.
- Langkah kelima : Lakukan kembali langkah ketiga sampai pemakai merasa puas.
- Langkah keenam : Bikin desain terakhir sebelumnya.

15.4 SISTEM YANG BERMANFAAT DARI BENTUK DASAR

Sejak syarat (baca FS) sangat diperhatikan dan menunjukkan sistem bentuk dasar kepemakai, hanya pemakai yang cukup memeriksa keluar persyaratan tersebut. Menu, bentuk gambar yang dimasukkan, bentuk yang keluar atau hasil cetaknya, aba-aba dan pesan adalah kandidat yang cocok untuk bentuk dasar. Dilain pihak, perhitungan yang lengkap, kelompok Update dan waktu yang benar adalah sistem ilmiah yang sangat susah untuk dijadikan model (kamu dapat menganggap dirimu untuk melakukan aktivitas, untuk contoh, print kamu, kelompok Update untuk transaksi dengan 10 transaksi baru yang biasanya sukses lengkap tanpa proses yang lain)

Sistem harus cocok untuk bentuk dasar yang satu yang bergantung pada pemakai input–output sistem dengan transaksi on-line dengan memakai menu, layar, bentuk, laporan query dan aba-aba.

15.5 SOFTWARE UNTUK BENTUK DASAR

Apa yang harus disediakan untuk paket software sebuah bentuk dasar

Produk yang baik harus disediakan ada 7 :

1. *Kreasi menu cepat dan mudah.* Menu harus dapat disebut sub menu, bentuk, laporan, program bentuk dasar dan persediaan garis untuk menolong menu seleksi dan cepat. Gambar 15.1 memperlihatkan konstruksi menu program (**hal 5**)
2. *Bentuk kreasi tampilan input dan output.* Kamu harus dapat ‘menggambar’ bentuk layar dari tempat kursor untuk menemukan lokasi (mouse melakukan yang terbaik untuk ini), tipe untuk field nama dan spesifikasi yang lain, memerlukan alphanumeric, nilai yang boleh dihitung, error dan pertolongan informasi dan banyak lagi. Gambar 15.2 memperlihatkan definisi dari program action (**hal 6**)
3. *Serupa / mirip.,* kamu harus bisa dengan mudah memperkenalkan format dari hasil laporan. Item-item menetapkan laporan generasi yaitu judul, catatan tambahan, dengan menaruhnya di sini (ini adalah program prompt terbaik dengan semua field yang diketahui), kolom pertama, grup, pemendekan dan sub dan total hasilnya. Kebiasaan satu yang bisa dijadikan laporan hanya dipilih oleh item saja. Gambar 15.3 menunjukkan penggunaan laporan program generator (**hal 7**)
4. Perangkat lunak harus menghasilkan secara otomatis sebuah kamus data (DD = data dictionary). Kamus data tersebut dapat menyimpan informasi dan mengenal masing-masing wujudnya seperti layar, laporan atau bentuk-bentuk; tetapi sangat penting, kamus data menyimpan informasi pada setiap field termasuk panjangnya sebuah field, menampilkan data yang dikehendaki dan dengan laporan-laporan serta bentuk-bentuk field yang digunakan. Kamus data adalah jantung yang menghasilkan dan sebagai mestinya suatu alat, alat prototipe dapat digunakan pada kamus data untuk memeriksa suatu field yang digunakan terdiri dari pada sebuah layar dan dapat melindungi pengetikan ulang jika sebuah field muncul lebih dari satu kali. Dan pada akhirnya, sebuah perangkat lunak dapat menyimpan jejak atau jalan dari semua hubungan diantara menu-menu, tampilan layar,

laporan-laporan dan data. (lihat bab 6 pada bagai seperti sebuah alat yang dapat digunakan bagi analisis yang baik dari prototipe).

5. Perangkat lunak haruslah mahir untuk membangun sebuah Database Rudy mentary. Definisi suatu layar peng-input / masukan seperti sebuah pengguna dalam figure 15.2 mengatakan suatu alat tentang bentuk sebuah record. Perangkat lunak harus dapat membentuk sebuah Database dan selanjutnya mengijinkan pemakai/pengguna untuk memasukkan data kedalam sebuah bentuk pemasukan data (input). Hasil yang baik memberikan pemakai/pengguna untuk Database yang optimis oleh penentu sebuah bentuk dan kunci dari suatu record.
6. Dilihat dari hasilnya dengan pertanyaan langsung yang cepat kerekaman data pada sebuah Database. Anda seharusnya mahir dalam melekukan penyelidikan yang sederhana, jenis , pilihan dan merekam sebuah tontonan/pameran.
7. Apa yang dibutuhkan didalam logika yang rumit atau menghitung suatu kebutuhan bagi prototipe? Meskipun tidak diperlukan, program yang baik memiliki bentuk yang sederhana dalam struktur bahasa pemrograman untuk mengizinkan anda melakukan proses khusus, waktu kejadian, prosedur yang otomatis dan sebagainya

PROTOTYPE SEBAGAI BAGIAN DARI CASE

Prototipe adalah sebuah metode dengan ketentuan yang bekerja sendiri/otomatis dan tahap dari suatu uraian, jadi ini bagian dari case. Tetapi jika langkah berikutnya untuk membuat sistem yang nyata didalam pemrograman didalam generasi ketiga, hasil prototipe yang menuntun kamu untuk mencetak semua keluaran, jenis itu mengetahui tentang semua menu (didalam permintaan dari struktur logika ketiga mereka). Semua bentuk dokumen dan perintah. Buatan terbaik mengikuti pengguna untuk mencetak sebuah logika mengatur dokumen yang khusus. Beberapa kejadian menyediakan kata-kata penggerak untuk memasukan kata keterangan diantara beberapa bagian. Seharusnya produk mencetak keluar kamus data (DD). Ini akan menyimpan sebuah rancangan banyaknya tenaga kerja. Jika sebuah rancangan pemrograman akan digunakan dalam

bahasa generasi keempat atau penggabungan perangkat case (lihat bagian 6.6) kamu seharusnya mahir dalam memasukan prototyping secara tepat kedalam sebuah perangkat. Perangkat case yang terbaik ini akan mengolah secara otomatis sebuah basis data dan kode yang terjadi pada aplikasi yang terakhir.

Berapa cepat kamu harus mengatasi bentuk dasar

Perangkat lunak prototipe harus mampu membuat kamu membangunnya dengan cepat model inisial dari sistem. Ciri, kamu bisa membangun prototipe pertama yang kecil untuk ukuran sedang hanya dalam waktu 2-3 minggu. Kamu harus mampu mengembangkan dengan cepat serta bagus. Itu mengambil beberapa menit untuk membuat beberapa kosmetik seperti menggerakkan sebuah bidang dalam sebuah contoh. Itu akan mengambil tidak lebih dari sejam untuk menentukan menu baru suatu contoh dan lebih dari beberapa jam untuk menciptakan file baru/mengadakan file baru untuk bidang baru untuk membuat field, key atau akses. Untuk menciptakan kecepatan ini diperlukan cara yang sederhana dan logika pengguna pengembangan, kenyataannya jika software prototipe sangat sederhana, pengembangan dapat melatih pengguna untuk menjalankannya pengguna kemudian akan mengembangkan prototipe dan memanggil pengembang jika itu telah siap untuk menjalankan kedalam sistem final.

15.6 SEMANGAT BENTUK DASAR DIBAGI 7 FASE.

Sekarang anda dapat merasakan metode prototipe itu bisa menyajikan secara utuh/komplit dapat diganti 7 tahapan datanya untuk perkembangan proyek C dan jika mungkin gangguan itu saya membuat kamu membaca sektor saya dari buku sebelum mendengarkan kamu tentang prototyping. Jangan takut prototyping dapat diganti hanya tempat dari definisi C tahapan . Gambar 15.4, urutan kejadian didalam tujuh tahapan dari metode lama sisi dengan pemberi informasi kejadian di dalam metode baru (prototipe).

METODE LAMA (7 BAGIAN)

METODE BARU (BENTUK DASAR)

Phase	activity	comment	activity	comment
--------------	-----------------	----------------	-----------------	----------------

Definistion	Req't Doc Proposal is guess	Had to Be estimate	Initial final req'ts]proposal	Truly initial Est.is accurate
Analysis	Initial func.Spc	Know it Will change	Not needed	Changes to the prototype as required
	Negotiate the FS		Not needed	
	Final FS		Final FS	Prot. Prints most of it. No negont'n needed prot.prints
Design	Des.spec	Req'ts Changes	Des spc. fewer changes	Prot. Prints DD, record, rest of design is same
Prog'g sys Tst acceptnc operat'n	Programs	Change	Programs	Same-less change
	Int. & test	Change	Int.&test	Same-less change
	Acc. Test	Change	Acc.test	No surprises- fast
	Cut-over, training		cut-over, training	No surprises- fast Less required

Gambar 15.4 bagaimana bentuk dasar dibagi menjadi tujuh bagian

Definisi dan analisa akan memakan waktu yang sedikit dari cara baru karena sebuah deadline akan ditentukan waktunya. Pernyataan ini tidak berlaku jika kamu menentukan deadline kamu dapat menentukan beberapa angka dari prototipe. Saya telah menentukan jika pengguna dan pembangun setuju untuk mempermudah berubah kesatu iterasi, sebuah prototipe dapat di bangun dengan menggunakan kurang dari 5 fersi, dengan 3 bagian yang sangat umum. Walaupun cara yang yang baru memakan waktu yang lama, kamu akan menghemat uang dan waktu dari seluruh proyek karena ini akan lebih sedikit mengalami perubahan. Tingkat disain akan lebih pendek, penerimaan akan dianggap tak berarti, dan dokumentasi pengguna dan training akan menjadi sepotong kue. Satu yang terakhir sebelum kamu melemparkan siklus hidup yang lama. Saya kira akan sia-sia jika kamu melakukan pekerjaan dengan cara yang lama terlebih dahulu, kamu tak akan benar-benar mengerti dan menggunakan metode prototipe. Ada sepatah kata yang harus dikatakan, “ kamu harus belajar mengendarai sebelum kamu mengendarai motor?”

15.7 MELIHAT BEBERAPA PRODUK

Bentuk dasar dari peralatan

Di bab 6 ini kita akan melihat analisa dari alat pemanggil mesin, tetapi alat ini mempunyai keistimewaan untuk membuat bentuk yang lebih besar dan bagus, Dan ini dapat membuat sebuah basis data yang berguna, dan bisa memasukan bentuk yang terpenting disini bisa mengizinkan anda untuk mencetak data dari sebuah logika yang mengatur fungsi perincian dokumen yang terdiri dari semua artikel/pokok masalah yang menegaskan dengan proses memasukan kata sebagai tambahan.

15.8 KESIMPULAN

Prototipe telah berhasil guna untuk mempertinggi dari implementasi dari berbagai sistem software yang melibatkan banyak penghubung dengan pengguna melalui menu, layar, laporan dan transaksi langsung. Sistem berorientasi bisnis yang istimewa sangat cocok untuk metode ini. Sistem real time dan sistem ilmiah muatannya akan sedikit berkurang ketika menggunakan metode ini.

Kemungkinan pertanyaan yang ada di pikiran anda dari point ini adalah, “mengapa mengikuti semua kesalahan untuk membangun sebuah sel kosong?”. Mengapa saya tidak memiliki software yang mengizinkan saya untuk membangun semua menu, tampilan layar, dan form dan berakhir dengan sistem kerja yang pasti? “Sesungguhnya kamu telah mengerjakan software itu. Baca bab berikutnya dari generasi bahasa ke-4.

Tingkat manajer tertinggi mampu mendapatkan posisi PM. Perhatikan sebuah organisasi, dan ia berkata “disekeliling kami banyak manajer dan salah satu dari mereka adalah PM!” hal ini bagus jika manajer-manajer ini mempunyai pernyataan dan keahlian khusus untuk mengatur sebuah proyek perangkat lunak (lihat bag 19.2). Kemampuan dari personal atau general manajer tidak sama seperti yang telah disyaratkan.

Organisasi, komunikasi dan pemimpin firma termasuk kedalam lingkungan multi person. Penjualan proyek manajemen sebagai sebuah pekerjaan yang di butuhkan. Periklanan sukses dari pengalaman pribadi dan dari industri atau katakan pada manajer untuk membaca buku ini.

TANGGUNG JAWAB UMUM DARI PROYEK MANAJER

Peraturan PM adalah saling bertemu antara kelompok proyek dengan dunia luas. Mereka bertanggung jawab untuk melaporkan rencana-rencana dan kemajuan klien, pada tingkat manajemen yang tertinggi dan pada seluruh perhatian, semua informasi dari perubahan luar, anggaran-anggaran, jadwal orang-orang, dan masalah perusahaan. Hal lainnya yang mempengaruhi PT adalah berkomunikasi kepada PM yang menginformasikan anggota kelompok. PM menempatkan seluruh persyaratan untuk mendapatkan pekerjaan.

Tanggung jawab lainnya adalah mengatur orang-orang. PM adalah pemimpin, mendorong dan memecahkan masalah orang yang membutuhkan. Dan terakhir tetapi bukan yang paling akhir, kelompok pertahanan PM berasal dari politik dan birokratik belaka yang berasal dari luar, (persyaratan untuk mendapatkan pekerjaan). Manajer proyek adalah penyangga atau layar untuk administrasi

TANGGUNG JAWAB KHUSUS PM

1. Definisi

PM mungkin salah satu yang membuat keputusan atau tidak membuat keputusan. Hal ini mungkin yang mempengaruhi pelaksanaan interview dan membantu penulisan dokumen persyaratan. PM kemungkinan yang membuat proposal, PL akan membantu dengan menyediakan asisten teknik beserta hal-hal lain seperti rencana persiapan proyek. PM akan merencanakan membuat laporan struktur pekerjaan dari tingkat yang tertinggi, kemudian WBS memberikan pada orang-orang teknik untuk menyelesaikan perubahan dan membuat anggaran belanja. Ketika perincian telah dilakukan maka PM akan mengkalkulasi harga keseluruhan, membuat jadwal dan membuat perjanjian untuk sumber penghasilan. PM akan melakukan segala ulasan dalam satu ulasan klien dan persetujuan, menegosiasikan proposal (surat permohonan), memastikan team tehnik menyetujui semua perubahan proposal dan menempatkan klien sesuai tempatnya.

Setelah proposal ditandatangani, PM secara formal, imitasi proyek dengan membangun Proyek Fice (PF). PF akan menjadi pusat pernapasan untuk semua versi terbaru dari semua proyek, termasuk laporan status, waktu rapat, yang memperhatikan proyek dan sebagainya. Ia menyebutnya pertemuan proyek, membantu peraturan dan membentuk perasaan yang antusias.

2. Analisis

PM harus menulis beberapa bagian dari spesifikasi dari fungsional (lihat bagian 6.3) dapat diantar, perubahan spesifikasi, penerimaan, pertemuan antara pelaksana dan team proyek, tanggung jawabnya (perincian berasal dari PL). Barang-barang dan

kondisi, jaminan dan segala sesuatu tentang pengaruh dari sistem baru pelaksana lingkungan.

PM akan menjamin bahwa FS (sistem dasar) melakukannya dengan tepat waktu (maksudnya bahwa pemasukkan adalah penerimaan pada waktunya), FS bernegosiasi dengan klien dan menempatkan klien sesuai dengan tempatnya. Bagian yang terberat adalah memastikan pemakai mengerti FS, bahwa perubahan tidak terjadi selamanya dan ada beberapa perubahan yang diakui oleh team tehnik

PM pada point ini mempunyai peranan mengurangi beban dengan merencanakan proyek. Ia membuat perjanjian yang berasal dari fungsional manajer-manajer untuk seluruh sumber penghasilan khususnya untuk perancang dan programer tentunya tidak tertutup untuk siapa saja yang membutuhkan latihan.

Permulaan kontrol pada situasi ini sudah bagus, PM melihat adanya kemajuan dengan kebutuhan status secara terus menerus dan adanya pertemuan berulang kali, penulisan status untuk seluruh perhatian dan dapat mengatasi masalah. Jika staf telah dipinjamkan oleh departemen lain maka penampilan individu harus ada timbal balik yang pantas pada manajer-manajer. Kontrol sangat penting ketika menggunakan bentuk dasar atau 4GL. PM memastikan bahwa model adalah mengimprofisasi, hal tersebut tidak banyak berhasil dan pengembang-pengembang harus memberikan dorongan yang tepat, dan ia dapat mengatasinya!. PM dapat meyakinkan bahwa berguna untuk dilibatkan dan dimotivasi.

3. Desain

Sejak keadaan teknik mulai tinggi, permintaan pekerjaan PM mulai berkurang. Prosedur kontrol secara formal kenyataannya sekarang mulai terbukti, dan membuat surat peringatan proyek. PM membuat pertemuan secara regural dan membuat laporan secara terus menerus. Ia mengecek waktu dan membuat anggaran setiap minggunya, ramalan harga akhir dan waktu pengiriman dan harapan pemasangan jika dibutuhkan. PM harus mendeteksi adanya masalah-masalah, mengatasinya jika mungkin, menanamkannya jika tidak. Jika ada pertemuan beberapa manajemen

seperti pertemuan panitia acara atau mengulang kembali kejadian yang penting, maka PM akan memimpin. Sangat penting, PM akan menjamin bahwa klien akan merasa puas dengan kemajuan produk.

Persyaratan mengalami perubahan pada waktu perencanaan, PM harus mendirikan dan melakukan perubahan prosedur kontrol untuk memastikan bahwa perubahan mempunyai pengaruh yang kuat didalam merefleksikan harga proyek.

4. Tahap ini melibatkan banyak dan masalahnya. Proyek manajer dapat memonitor reaksi team dan kegembiraan secepat mungkin jika sesuatu terjadi kesalahan. Manajer yang terkait, sehingga PM dapat memperhatikan masing-masing keberadaan penampilan. Tahap ini dapat memakan waktu yang lama tanpa suatu peringatan yang jelas. PM dapat menjamin dengan baik kemajuan yang dibuatnya, PM harus jalan mengelilingi skemanya, berbicara kepada programer dan menggunakan imajinasi untuk mendeteksi beberapa masalah.

5. Keutuhan sistem dan mengujinya

PM terbaik dapat melakukan di fase ini untuk menyimpan keutuhan rambut. Di dunia luar (anggaran, level manajemen yang tinggi) biasanya khawatir sekarang, hubungi PM, memanggil dan bertemu dan bertanya tentang laporan kemajuan setiap hari. Ini ketika PM sudah siap untuk diperlukan menyimpan jauh dari PT dan yang terakhir laporan PL yang ada disemua sistem berusaha belanja, PM pertama kali menjamin ATP lari dan mengoreksi dan memanggil mayor millestone untuk bertemu (pizza dan bir) untuk mengumumkan keseluruhan dunia bahwa “ kami mempunyai sistem”.

Part 3

People

BAB 18

ORGANIZATION

Who Does What and When

18.1 INTRODUCTION

Kita semua pernah menghadiri rapat dimana semua jenis tugas didiskusikan tetapi tak satu pun dari satu dari kita menyadari pentingnya hal tersebut, yang tak pernah berjalan dengan baik karena semua orang berpikir bahwa orang lain lah yang bertanggung jawab akan hal tersebut. Tujuan utama sebuah proyek yang besar akan tercapai jika semua orang menyadari tanggungjawabnya masing-masing.

18.2 ORGANIZING THE PROJECT TEAM

The Small to Medium Sized Project Team

Seperti yang telah dijelaskan pada Chapter 3 (Planning), suatu tim yang baik dari sebuah proyek kecil hingga menengah adalah seperti yang digambarkan dalam **Figure 18.1**.

Semua orang dalam tim mempunyai tugas khusus masing-masing. Semua orang-orang teknik memerlukan pengawasan teknik, itu sebabnya **Project Leader** (PL) dibutuhkan untuk memimpin semua aktifitas yang berhubungan dengan teknik dan menyelesaikan semua masalah-masalah yang berhubungan dengan sistem. Tanggungjawab seorang **PL** adalah kualitas dari produk yang dihasilkan. **Project Manager** (PM) dibutuhkan untuk menjalankan semua manajemen dan handle semua komunikasi antara **Project Team** dan dunia luar. Tanggungjawab utama seorang **PM** adalah untuk merencanakan dan mengawasi.

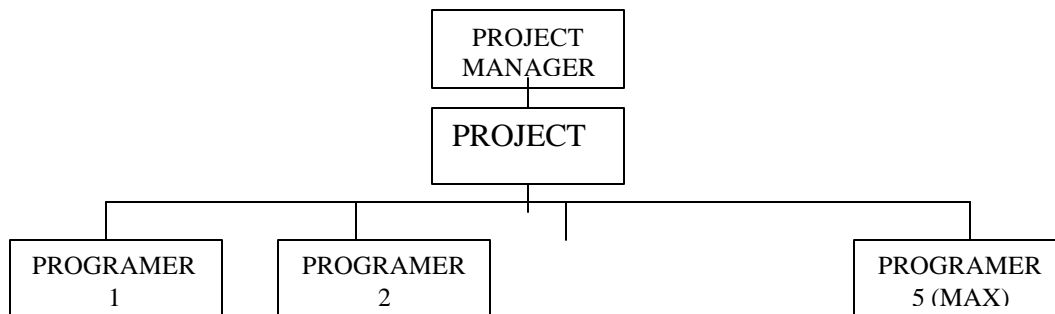


Figure 18.1 Project Team

The Larger Project Team

Dalam sebuah Project Team yang besar seperti pada bagan pada **Figure 18.2**, seorang **PL** tidak dapat mengawasi lebih dari lima programmer secara efektif.

Jika anda menjalankan organisasi seperti ini, bagilah **Project team** besar tersebut sedemikian rupa sehingga para individual didalamnya serasa menjalankan

suatu proyek yang berdiri sendiri. Salah satu konklusi yang menarik dibuat oleh **Tom Peters** dalam bukunya *In Search of Excellence* adalah Produk terbaik didunia diproduksi oleh sebuah tim yang terdiri tidak lebih dari tujuh orang.

The Functional Project Organization

Kebanyakan perusahaan telah terorganisir secara fungsional. Sebagai contoh, jika suatu perusahaan software bertanggungjawab atas beberapa jenis software, pasti ada bagian dari perusahaan yang bertugas memproduksi jenis software tersebut. Software perbankan dikerjakan oleh bagian aplikasi perbankan, aplikasi pengontrolan proses dikerjakan oleh bagian penotrolan proses, dan begitu seterusnya. Manajer dari bagian aplikasi perbankan yang disebut sebagai manajer fungsional, otomatis menjadi **PM** untuk pembuatan software perbankan, dan manajer fungsional dari bagian proses adalah **PM** untuk bagian aplikasi pengontrolan proses.

Mengorganisir dengan cara ini mempunyai keuntungan dan kerugiannya. Keuntungannya adalah anda dapat dengan gampang menemukan bagian dan manajernya untuk sebuah proyek yang familiar. Kerugiannya adalah proyek-proyek biasanya membutuhkan seorang ahli dari luar group, dan masalah dapat terjadi apabila kita harus meminjam ahli dari group lain. Ahli ini biasanya dipinjamkan secara part time untuk proyek tersebut, dan kita dapat melihat apa yang terjadi apabila seseorang harus membagi waktu mereka untuk beberapa proyek yang berbeda, mereka lebih sibuk mengatur waktu daripada memproduksi. Dan akan membosankan apabila selalu bekerja dengan orang yang sama, manajer yang sama, dan dalam sebuah proyek yang sama.

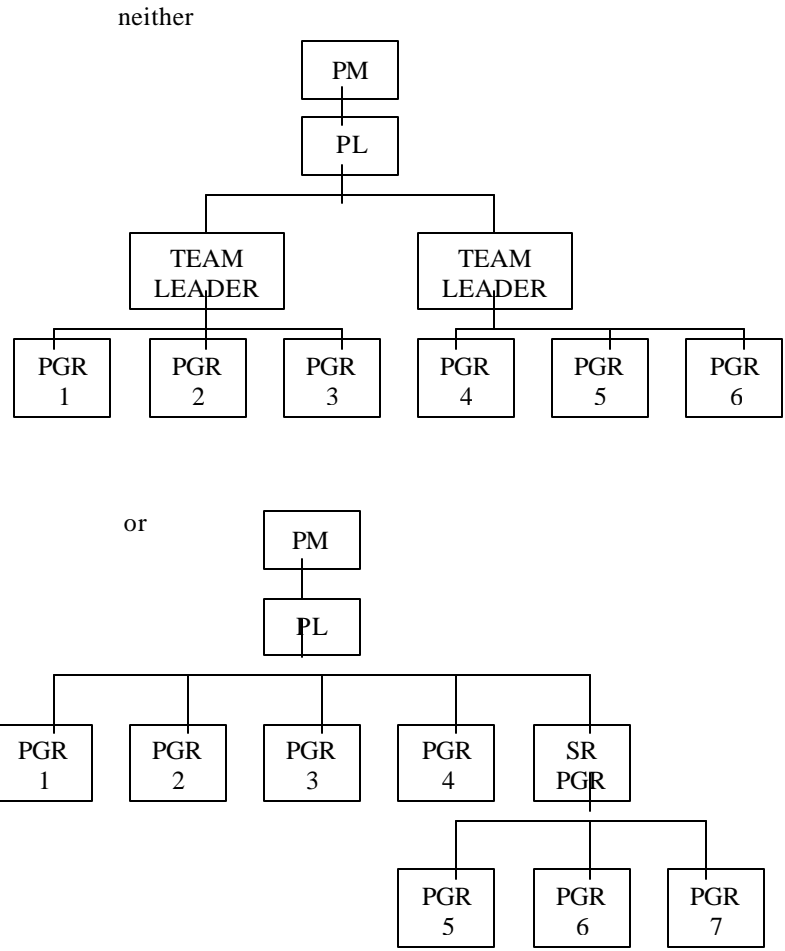


Figure 18.2 Larger Project Teams

The Matrix Project Organization

DEC dan perusahaan penghasil software lainnya kadang-kala menggunakan metode dari organizing project team; beberapa kelompok dari para programmer melapor kepada **Line Manager** (LM). Para programmer ada dalam kelompok khususnya hanya karena mereka selalu ada dalam kantor yang sesuai dengan keahliannya atau pekerjaan yang sesuai dengan keahliannya ketika mereka disewa. Ketika sebuah proyek dimulai, seseorang dengan keahlian yang sesuai adalah yang pertama dicari

untuk menjalankan proyek tersebut. **PM** ini biasanya adalah seorang programmer yang sudah senior, yang mempunyai banyak pengalaman dengan klien dan aplikasinya, dan yang tahu bagaimana menghadapi orang. **PM** ini lalu menentukan programmer mana yang terbaik untuk membangun proyek yang bersangkutan dan bernegosiasi dengan **Line Manager** masing-masing untuk menggunakan programmer tersebut.

Orang-orang yang telah diangkat bekerja fulltime, dan pada akhir proyek orang-orang yang bekerja tersebut kembali ke **LM**-nya masing-masing. Lalu **PM** membayar **LM** untuk penggunaan para programmer tadi, dan pembayaran disesuaikan dengan keuntungan yang didapat dari proyek tersebut. **PM** dan **LM** mempunyai andil yang besar atas suksesnya suatu proyek.

Bentuk dari **Matrix Management** ini dapat bekerja dengan baik apabila **PM** dan **LM** mempunyai tanggungjawab dan wewenang yang sama. Secara tidak langsung mereka berdua mempunyai suara yang sama dalam keterlibatannya untuk menentukan suatu keputusan. Di beberapa perusahaan, kadang manajer yang satu kedudukannya lebih tinggi dari manajer lainnya. Contoh dalam DEC, karena **LM** adalah pemimpin yang permanen sedangkan **PM** adalah pemimpin sementara, maka **LM** adalah pemimpin yang paling punya wewenang atas para programmer.

Meskipun begitu mengorganisir sebuah team secara dinamik untuk menyelesaikan suatu masalah adalah baik sekali. **Tom Peters** menemukan bahwa produk yang baik dihasilkan dari sebuah tim yang terorganisir secara dinamis. Orang termotifasi untuk bekerja dengan manajer dan anggota tim yang berbeda dan masalah-masalah yang berbeda, selama tidak terlalu sering.

Reorganization gives the illusion of progress. In reality, it creates demoralization and insecurity.

Petronius (Roman General) 66 A.D.

18.3 THE ROLE OF PROJECT MANAGER

Selling the Project Manager Job

Jika anda menjalankan sebuah Project team seperti yang digambarkan pada Figure 18.1, maka tidak akan ditemukan masalah dalam mempercayakan seseorang untuk diangkat sebagai **Project Leader** (semua Technical Teams membutuhkan pengawasan), tetapi anda akan menemukan masalah dalam menemukan seorang **Project Manager**. Karena tugas utama seorang **PM** adalah menghadapi pemakai, maka para pemakai tersebut akan memperdebatkan , “Kenapa menugaskan seseorang untuk berkomunikasi dengan saya? Saya akan “mengatur” proyek anda!” Hal ini mungkin terjadi dalam suatu proyek internal. Tapi masih lebih banyak tugas untuk mengatur sebuah proyek daripada harus menghadapi para pemakai. Apakah para pemakai mengerti bagaimana meghandel orang, menghadapi atasan, mendapatkan sumberdaya, dan mengontrol suatu kejadian yang penting? Akankah pemakai ini menerima apa adanya ketika ada permintaan dari pihak pemakai untuk melakukan perubahan penting, sedangkan **Project Team** tidak mau mengerjakannya?

18.4 Aturan dari pimpinan proyek

Pertanggung jawaban umum dari pimpinan proyek (PL)

Tujuan utama dari pimpinan proyek adalah menghasilkan produk berkualitas tinggi. Padahal manajer proyek berorientasi kedalam proyek tersebut. Jika kamu adalah pimpinan proyek kamu akan memastikan bahwa produk yang dibuat sesuai dengan spesifikasinya dan tidak akan dipenuhi oleh virus. Kamu akan memimpin seluruh aktifitas tehnik, memimpin setiap pemeriksaan ulangan secara tehnik, memberikan tugas kepada disainer dan programer, memecahkan permasalahan utama dan mungkin mengerjakan semua tugas yang kompleks sendiri.

Pertanggung jawaban istimewa dari pimpinan proyek dalam setiap tingkatan

1. Definisi

Pimpinan proyek akan mengerjakan seluruh rencana pendahuluan proyek. Setiap WBS pada level yang rendah dan penaksiran (atau setidaknya pengawasan terhadap para penaksir) jika pemakai harus melengkapi seluruh teknik secara detail pada point ini atau jika ada model pertama yang dilibatkan, pimpinan proyek akan menjadi kepala interface.

2. Analisa

Pada proyek yang sederhana pimpinan proyek adalah pimpinan analisis dan menulis seluruh teknik spesifikasi fungsi dari setiap seksi seperti pada overfleu, obyektif, sistem persyaratan dan komponen deskripsi (lihat pada seksi 6.3) jika manajer proyek memerlukan asisten teknik dengan FS negoisasi atau perlu mengetahui setiap beban harga, pimpinan ahli teknik berada pada perkiraan sampingan siap membantu. Pimpinan proyek harus juga membantu memilih para anggota dari team proyek.

3. Desain

Pimpinan proyek adalah pimpinan dari team desain mereka akan memimpin seluruh pemeriksaan desain dan menjalankannya, memberikan tugas kepada desainer dan mungkin melakukan seluruh pekerjaan desain, khususnya pada level yang tinggi.

4. Pemrograman

Inilah saat pimpinan proyek menggunakan kaos dengan ukuran "S" yang besar pada dada. Pimpinan proyek akan membuat seluruh penyerahan program pada setiap pembuat program dan memecahkan setiap permasalahan yang programmer tidak dapat pecahkan. Dia harus membuktikan program desain, rencana test, kode aturan dan pemakai dokumentasi dan menjamin para programmer berinteraksi jika di perlukan. Program yang panjang atau program yang kompleks dapat diserahkan pada programmer yang menjadi asisten dari pimpinan proyek, ini membuktikan pimpinan proyek mengkode seksi kompleks ketika programmer proyek mengkode seksi kompleks ketika

para programmer mengisi secara detail. Pimpinan proyek dapat juga menulis seluruh program, tetapi tidak pada kolom kritikal.

5. Sistem test

Integrasi dan ujian akhir pada produk adalah tanggung jawab. Secara keseluruhan dari pimpinan proyek dia merencanakan integrasi. Mengontrolnya, menetapkan hasil pada jalurnya dan menetapkan manajer proyek menginformasikan setiap perkembangan mungkin harian.

6. Penerimaan

Pimpinan proyek akan menjalankan aspek tehnik dari penerimaan, dia mungkin akan menulis ATP, melaksanakan penyelesaian pekerjaan pada saat penyelesaian test sistem dan melakukan penerimaan kepada pemakai.

7. Operasi

Walaupun programmer senior cukup memadai, pimpinan proyek akan memberikan jaminan pekerjaan perorangan atau seseorang yang mampu dengan melalui telepon menjawab pertanyaan atau sama dalam menjalankan latihan bagi seluruh pemakai.

18.5 Pemakaian beberapa HATS pada saat bersamaan.

Jika anda adalah lima orang pada suatu perusahaan atau departemen, orang yang sama mungkin seorang manajer proyek, pimpinan dan juga para programmer proyek-proyek kecil. Kamu akan mendapatkan pilihan yang kecil disini, tetapi aku sarankan bahwa kamu mencoba memisahkan pekerjaan administrasi (PM) dari pekerjaan tehnik (PL), ada beberapa alasan untuk ini. Pertama jika pimpinan proyek mengawasi empat atau lima programmer (dalam melakukan analisa, desain, pemrograman ada penyelesaian masalah didalam), dia akan mendapatkan 100% waktunya digunakan. Proyek manajemen dari proyek kecil adalah tidak 100% (industri standar, industri tetapkan bahwa memajemen proyek menggunakan 15 % dari seluruh pekerjaan). Tetapi ini adalah kejadian pada pekerjaan perjalanan, kamu secara konstan terpengaruhi untuk menghadiri rapat, jawab setiap telepon dan pemadam api. Mengerjakan kerja waktu paruh sepanjang

dengan tidak mengikuti tata cara kerja membuat kedua pekerjaan tersebut sangat baik untuk menempatkan “fire chief” ditempat penanganan administrasi dan interupsi.

Alasan kedua untuk memisahkan pekerjaan PL dan PM ialah secara kejiwaan diantara definisi dan analisis pengguna dan PL berdiskusi bersama membicarakan menghasilkan sesuatu yang baru dan menjadi teman yang baik keberanian pemakai melakukan perubahan. Tetapi permulaan dengan tingkatan desain pemakai tidak mampu melakukan perubahan atau penolakan secara seketika. Jika ini adalah PL yang mengatakan tidak terhadap perubahan ini, pemakai akan merasakan bahwa perubahan teman yang tiba-tiba menjadi musuh, dan ini akan memanjakan hubungan yang diperlukan selama proyek berjalan. Jika terjadi pemisahan PM dan perubahan proses pengontrolan di tempat, pemakai tidak membantah perubahan tersebut. PL dapat dengan mudah mengingatkan pemakai bahwa seluruh perubahan harus melalui saluran yang khusus dan setiap orang dapat mengingatkan teman baik.

18.6 Disuatu waktu beberapa proyek berjalan

Jika proyek menejer menduga sejumlah pengeluaran hanya 15% dari waktunya pada proyek, bagaimana satu kemungkinan membayar pekerjaan penuh seseorang untuk memanejemen suatu proyek-proyek solusinya adalah bagaimana memberikan kepada PM beberapa proyek untuk diatur pada saat bersamaan, hati-hati pikirkan. Sama dengan tingkatan pada tempat proyek memiliki perbedaan pelaksanaan dengan PM. Seperti pada contoh. 1.12 Contoh tingkatan pertama dan dua contoh tingkatan terakhir menentukan bahwa prioyek akan ditangani secara lebih oleh PM waktu.

Tiga tingkatan pusat (desain, pemrograman, sistem ujian) akan menerima sedikit. Karena biasanya diambil sepanjang waktu. PM yang baik dapat mengatur banyak lima atau enam proyek tersebut. Disatu waktu, selama itu tidak membuang waktu proyek dan fase bersama-sama.

18.7 Aturan programmer

Seorang programmer menerima rancangan tingkat medium untuk modul rancangan tingkat terendah, rencana bagian mengetes suatu modul, memberi kode-kode, mendokumentasikan, dan kemudian mengetesnya. Seorang programmer bisa juga bertanggung jawab menggunakan pelatihan dan dokumentasi, walaupun dapat menyewa penulis profesional dan dapat mengerjakan semuanya. Programmer yang baik tidak perlu seorang penulis yang baik atau guru. Seorang programmer melaporkan perkembangan kepada PL, dan membantu pada waktu sistem dengan lengkap dan menguji aspek pada sistem yang mencakup program-program mereka. Hati-hati dengan persahabatan antara pengguna dan programmer. Skenario diatas sangat umum, terutama dalam proyek internal.

Act I. Scene 1 (fase program, pengguna berbicara dengan PM)

Pengguna:

PM : Sepertinya sulit, tapi saya akan mempertimbangkannya pada PT.

Act I. Scene 2 (rapat selanjutnya, PM tidak mempertimbangkan perubahan pada PT, karena semua orang menyerah terlebih dahulu, proyek sudah terlambat dan kelebihan anggaran belanja dan PM akan disalahkan)

PM : Perubahan itu akan merugikan anda 25% dari anggaran belanja dan dalam enam bulan jadwal akan berubah.

Pengguna : Saya kira tidak.

Act II.scene 1 (perubahan prosedur, pengguna mengundang programmer kesebuah bar, setelah beberapa kali minum.....)

Pengguna : Berapa lama waktu yang dibutuhkan untuk menambah bidang X untuk laporan Y

PGR : (bersendawa) tanpa berkeringat. Beberapa jam kemudian mempertimbangkan yang telah dikerjakan

Act II.scene 2 : Pengguna bertemu dengan PM

Pengguna : Saya ingin bidang X ditambah kelaporan Y

PM : Anda lihat terakhir kali bagaimana sulitnya menambah bidang-bidang itu kelaporan Y.

Pengguna : Oh yaa....!!! Programmer anda berkata : itu akan memakan waktu beberapa jam saja

Alternatif : Act II. Scene 2 : (penerimaan waktu)

PM: Dari mana bidang-bidang asal laporan Y

User dan PGR : OH..... apakah kami tidak memberitahunya padamu ?

PL : Apakah anda menyadarinya dan menempatkan sesuatu pada formulir Z di luar keseimbangan.

Komentar : Programmer menerima pengguna selama tak ada ikatan.

18.8 Pertanggung jawaban PM, PL, dan Programmer dalam lingkungan 4GL

Susunan kelompok menyarankan dalam bentuk 18.1 akan bekerja sebaik dalam lingkungan 4GL dengan satu perubahan besar. Kelompok ini dapat diperkecil tiga atau empat orang sudah cukup untuk memprogram aplikasi kecil. Aturan-aturan ini akan berubah sedikit lebih baik.

Aturan manajer proyek

PM masih bertanggung jawab pada administrasi proyek mereka akan selalu mengadakan pertemuan pengulangan, bentuk laporan dan tanda tangan tanggung jawab. PM akan menjadi lebih ahli dari sebelumnya. Saat team menjadi lebih kecil PM lebih dapat menguasai. Mereka akan lebih terbiasa dengan 4GL dengan menggunakan bahasa yang lebih baik. Pengetahuan ini dibutuhkan karena PM dalam aturan kepala negosiasi harus meyakinkan pengguna untuk kompromi dalam kesempatan tertentu untuk mendapat keuntungan dalam menggunakan 4GL. Seorang PM harus dapat mengajar tentang 4GL dan menentukan keuntungan : apa yang dapat diharapkan pada model

pertama berapa banyak yang didapat. Apa yang akan dilakukan dan di kerjakan dalam waktu dekat.

Peranan dari pimpinan proyek

PL akan menjadi kepala pembangunan dari 4GL. Seperti melakukan keperluan-keperluan dan menganalisa data, mereka akan menjadi pekerja tunggal yang lebih dekat dengan pengguna untuk membangun prototype dan pengurangan. PL akan membutuhkan kemampuan baru : komunikasi untuk menjelaskan produk, untuk membantu membuat keperluan-keperluan dan untuk membantu PM meyakinkan pengguna untuk berkompromi.

Peranan dari pembangun

“ Hacker” klasik (membuat kode sendiri didalam sebuah sudut, tidak berbicara dengan siapa-siapa, memakai sebuah kaos oblong kotor) untuk waktu terpanjang dalam daftar jenis berbahaya, sekarang mendekati kepunahan. Pembangun 4GL sekarang adalah seorang komunikator cakap yang kelihatan berkelakuan baik, karena mereka harus bertemu langsung dengan pengguna.

18.9 Tingkatan dari lini atau fungsi manajer

Fungsional atau manajer ini meminjamkan karyawan mereka keluar dari proyek, dan akan mendapatkan mereka kembali ketika proyek itu akan berakhir. Sejak FM bertanggung jawab untuk kesenangan, pertumbuhan dan motivasi dari manusia, ia harus menjaga keutuhan dari proyek dimana mereka terlibat. FM harus memastikan bahwa orang-orang melakukan perjanjian dengan adil, menikmati proyek itu dan tidak terkubur didalamnya. FM akan memperhatikan beberapa rapat proyek, bertemu sekali-kali dengan orang-orang da, mengirimkan laporan status.

Sebagai organisasi yang paling besar, tujuan dari FM mungkin mengalami benturan waktu dengan tujuan PM. Untuk satu proyek FM hanya sebagian dari semua bisnis, tetapi untuk PM satu proyek mungkin satu-satunya menjadi tanggung jawabnya. Level paling bawah dari manajemen harus memastikan bahwa tanggung jawab dan kekuasaan dari FM dan PM harus benar-benar ada komunikasi dan persetujuan dari kedua belah pihak.

18.10 Siklus pengguna

Pengguna mungkin memiliki sedikit tanggung jawab seperti menulis dokumen atau melengkapi tes data. Dalam beberapa masalah seorang pengguna harus mengadakan perjanjian dengan koordinator proyek untuk berhadapan dengan team proyek dari manajemen seorang klien juga harus memastikan bahwa disana sedikitnya ada orang yang mengerti dan dapat menjawab pertanyaan tehnik dari PT.

Pengguna harus mengambil usaha untuk mempelajari tentang manajemen proyek jadi mereka mengetahui saat ada pembicaraan dan menandatangani itu semua. Ketika ada dokumen yang dipercaya, pengguna harus membaca dan mengembalikannya tepat waktu. Jika pengguna ketinggalan pada saat tingkat (fase) tehnik, gunakan PM sebagai kontak sendiri. Berpegangan pada peraturan seperti mengubah prosedur, mereka mungkin menemukan bahwa PT akan menemukan produk yang baik.

18.10 Kesimpulan

Anda dapat menemukan dan memberikan tanggung jawab untuk setiap tugas yang berbeda. Tetapi apakah tugas yang dibuat itu sanggup dikerjakan. Siapa anda? Saya percaya anda dapat melaksanakan tanggung jawab ini dengan baik. Tanggung jawab untuk aktifitas pemanggilan atau komunikasi keinstansi baru. Tanpa ada hambatan yang berarti, lalu tanggung jawab itu akan terlaksana semestinya tanpa ada hambatan.