

The background of the cover is a photograph of a desk setup. A laptop is open, with a tablet resting on it. The tablet screen shows a video player interface with a play button. In the foreground, there are several sheets of paper with diagrams and flowcharts. The entire scene is overlaid with a semi-transparent pink and purple gradient.

Manajemen Proyek Perangkat Lunak

A decorative graphic consisting of three slanted, parallel bars in shades of pink and purple.

Teori & Implementasi



Suyono, Mardiyanto, Jarudin, Tri Susilowati,
Akrim Teguh Suseno, Linda Suvi Rahmawati,
Sukenada Andisana, Frhendy Aghata,
Ibnu Atho'llah, Weda Adistianaya Dewa

Manajemen Proyek Perangkat Lunak

(Teori dan Implementasi)

**Suyono, Mardiyanto, Jarudin, Tri Susilowati, Akrim
Teguh Suseno, Linda Suvi Rahmawati, Sukenada
Andisana, Frhendy Aghata, Ibnu Atho'illah, Weda
Adistianaya Dewa**



PT. MIFANDI MANDIRI DIGITAL

Manajemen Proyek Perangkat Lunak

(Teori dan Implementasi)

Suyono, Mardiyanto, Jarudin, Tri Susilowati, Akrim Teguh Suseno, Linda Suvi Rahmawati, Sukenada Andisana, Frhendy Aghata, Ibnu Atho'illah, Weda Adistianaya Dewa

ISBN: 978-623-88663-6-6

Editor : Sarwandi

Penyunting : Miftahul Jannah

Desain sampul : Rifki Ramadhan

Penerbit

PT. Mifandi Mandiri Digital

Redaksi

Komplek Senda Residence Jl. Payanibung Ujung D
Dalu Sepuluh-B Tanjung Morawa Kab. Deli Serdang
Sumatera Utara

Distributor Tunggal

PT. Mifandi Mandiri Digital

Komplek Senda Residence Jl. Payanibung Ujung D Dalu
Sepuluh-B Tanjung Morawa Kab. Deli Serdang Sumatera
Utara

Cetakan Pertama, Agustus 2023

Hak Cipta © 2023 by PT. Mifandi Mandiri Digital

Hak cipta Dilindungi Undang-Undang

Dilarang memperbanyak karya tulis ini dalam bentuk dan
dengan cara apapun tanpa ijin tertulis dari penerbit

Kata Pengantar

Selamat datang dalam buku "Manajemen Proyek Perangkat Lunak". Buku ini merupakan panduan komprehensif yang dirancang untuk memberikan wawasan mendalam tentang praktik-praktik terbaik dalam mengelola proyek pengembangan perangkat lunak dengan sukses. Dalam dunia yang semakin tergantung pada perangkat lunak, kemampuan untuk mengelola proyek dengan efisien dan efektif menjadi semakin penting.

Pengembangan perangkat lunak adalah proses kompleks yang melibatkan banyak aspek, termasuk perencanaan, pengorganisasian tim, pengawasan, dan penilaian risiko. Buku ini akan membantu Anda memahami fondasi manajemen proyek dan memberikan panduan langkah demi langkah untuk mengatasi tantangan yang mungkin muncul selama perjalanan pengembangan perangkat lunak.

Dalam bab-bab awal, kita akan menjelajahi konsep dasar manajemen proyek, mengapa itu penting dalam konteks pengembangan perangkat lunak, dan bagaimana mengidentifikasi tujuan proyek yang jelas. Kami juga akan membahas berbagai pendekatan metodologi, mulai dari model air terjun hingga pendekatan agile yang fleksibel, yang telah membuktikan kesuksesannya dalam berbagai lingkungan pengembangan.

Kami yakin buku "Manajemen Proyek Perangkat Lunak" ini akan menjadi panduan berharga bagi para profesional di bidang teknologi informasi, pengembang perangkat lunak, dan siapa saja yang terlibat dalam pengelolaan proyek. Dengan penekanan pada aplikasi praktis dan contoh nyata, buku ini membantu Anda memahami tidak hanya teori tetapi juga penerapan nyata dari prinsip-prinsip manajemen proyek yang sukses.

Terima kasih telah memilih buku ini sebagai sumber wawasan Anda tentang manajemen proyek perangkat lunak.

Kami berharap buku ini membantu Anda mengatasi tantangan dan mencapai keberhasilan dalam setiap proyek yang Anda jalankan.

Medan, Juli 2023

Penulis

Daftar Isi

Kata Pengantar	i
Daftar Isi	iii
Bab 1 Konsep Dasar Manajemen Perangkat Lunak	1
Pendahuluan	1
People	3
Problem	6
Proses	12
Bab 2 Konsep Perangkat Lunak	19
Pendahuluan	19
Definisi Perangkat Lunak	19
Jenis-jenis Perangkat Lunak	20
Proses Pengembangan Perangkat Lunak	25
Metodologi Pengembangan Perangkat Lunak	27
Manajemen Konfigurasi Perangkat Lunak	29
Lisensi Perangkat Lunak	31
Bab 3 Kerangka Kerja Manajemen Proyek Perangkat Lunak	34
Pendahuluan	34
Manajemen Proyek Perangkat Lunak	36
Produk	46
Proses	47
Proyek	49
Bab 4 Teknik Estimasi Biaya	52
Pendahuluan	52
Estimasi Berbasis Analogi	53
Estimasi Berbasis Bottom-Up	54
Estimasi Berbasis Parametrik	56
Estimasi Berbasis Three-Point	57
Estimasi Berbasis Top-Down	59
Estimasi Berbasis Expert Judgment	60
Estimasi Berbasis Pemilihan Vendor	61
Bab 5 Manajemen Waktu Proyek Perangkat Lunak	64
Pendahuluan	64

Tujuan Manajemen Waktu	65
Mekanisme Manajemen Waktu	65
Bab 6 Manajemen Mutu Proyek Perangkat Lunak	78
Pendahuluan	78
Manajemen Mutu Proyek Perangkat Lunak	79
Tahapan Manajemen Mutu	80
Bab 7 Manajemen Resiko Proyek Perangkat Lunak	100
Pendahuluan	100
Perencanaan Manajemen Risiko	101
Identifikasi Risiko	102
Analisis Risiko	105
Perencanaan Penanganan Risiko	107
Pemantauan Risiko	109
Bab 8 Dokumen Kontrak	111
Pendahuluan	111
Surat Perjanjian	112
Surat Penawaran	116
Surat Pengadaan Barang Penunjang Aplikasi	118
Surat Perintah Kerja	119
Bab 9 Organisasi Dan Personel Proyek Perangkat Lunak	120
Pendahuluan	120
Struktur Organisasi Proyek Perangkat Lunak	121
Personel Proyek Perangkat Lunak	129
Bab 10 Pengendalian Proyek	136
Pendahuluan	136
Tujuan Pengendalian Proyek	136
Mekanisme Pengendalian Proyek	137
Pemantauan Kinerja Proyek	137
Pengendalian Jadwal	138
Pengendalian Biaya	140
Gantt Chart	141
Metode Jalur Kritis/Critical Path Method (CPM)	141
Metode Rantai Kritis (CCM)	142
Manajemen Nilai yang Diperoleh (EVM)	142
Pengelolaan Pemangku Kepentingan	143
Pengelolaan Persyaratan	143
Pengelolaan Perubahan Proyek	144

Pengelolaan Risiko	144
Pengelolaan Masalah dan Keputusan	145
Pengelolaan Kualitas	145
Pengelolaan Penerimaan Hasil Kerja	146
Pengelolaan Transisi	146
Pengelolaan Implementasi Bisnis	146
Pengelolaan Alih Daya	147
Daftar Pustaka	148
Tentang Penulis	156

BAB 1 KONSEP DASAR MANAJEMEN PERANGKAT LUNAK

Pendahuluan

Manajemen Perangkat Lunak (*Software Management*) adalah proses perencanaan, pengorganisasian, pengkoordinasian, dan pengendalian sumber daya manusia dan teknis yang terlibat dalam pengembangan dan pemeliharaan perangkat lunak. Konsep dasar manajemen perangkat lunak ini penting untuk memastikan proyek perangkat lunak dapat berjalan dengan efisien dan efektif serta menghasilkan produk perangkat lunak yang berkualitas tinggi.

Berikut adalah beberapa konsep utama dalam manajemen perangkat lunak:

1. Perencanaan Proyek
Ini melibatkan definisi tujuan proyek, identifikasi persyaratan, alokasi sumber daya, dan penentuan jadwal. Perencanaan yang baik membantu dalam menghindari kebingungan dan ketidakpastian selama proyek berlangsung.
2. Pengorganisasian Tim
Memilih anggota tim yang sesuai untuk peran tertentu, menetapkan tanggung jawab, dan membangun struktur organisasi yang efektif adalah bagian penting dari manajemen perangkat lunak.
3. Pengendalian Versi
Ini mencakup manajemen kode sumber dan dokumentasi proyek. Sistem pengendalian versi

membantu dalam melacak perubahan, mengelola cabang pengembangan, dan menghindari konflik dalam kode yang dikelola oleh beberapa pengembang.

4. Pengendalian Risiko

Mengidentifikasi, menganalisis, dan mengelola risiko yang mungkin timbul selama pengembangan perangkat lunak. Tindakan pencegahan dan rencana cadangan dirancang untuk mengatasi risiko ini.

5. Estimasi Biaya dan Waktu

Penentuan biaya dan waktu yang diperlukan untuk menyelesaikan proyek adalah esensial untuk mengelola anggaran dan menetapkan harapan yang realistis bagi para pemangku kepentingan.

6. Pengujian dan Validasi

Tahap pengujian perangkat lunak untuk memastikan bahwa itu berfungsi sesuai dengan persyaratan dan ekspektasi yang telah ditetapkan. Ini melibatkan pengujian fungsionalitas, kinerja, keamanan, dan kualitas lainnya.

7. Manajemen Konfigurasi

Ini adalah tentang mengelola komponen perangkat lunak yang berubah selama siklus hidup proyek. Ini meliputi dokumentasi, pelacakan perubahan, dan mengelola versi yang berbeda dari perangkat lunak.

8. Manajemen Perubahan

Ketika ada perubahan dalam persyaratan atau arah proyek, manajemen perubahan memastikan bahwa perubahan ini dikelola dengan baik dan memiliki dampak minimal pada proyek.

9. Komunikasi dan Kolaborasi

Komunikasi yang efektif antara anggota tim dan pemangku kepentingan memfasilitasi pemahaman yang jelas tentang proyek, kemajuan, dan masalah

yang mungkin timbul.

10. Pemantauan dan Evaluasi

Pemantauan proyek secara teratur membantu dalam mengukur kemajuan, mengidentifikasi keterlambatan atau masalah, dan mengambil tindakan korektif jika diperlukan.

11. Pengiriman Produk

Ini melibatkan pengiriman produk perangkat lunak yang sudah siap kepada pengguna atau klien. Pengiriman harus dipersiapkan dengan baik dan melibatkan pengujian akhir serta pelatihan bagi pengguna.

12. Pemeliharaan dan Dukungan

Setelah pengiriman, perangkat lunak memerlukan pemeliharaan rutin, peningkatan, dan dukungan untuk memastikan berfungsinya yang baik dalam jangka panjang.

Manajemen perangkat lunak adalah proses yang dinamis dan terus berubah sepanjang siklus hidup pengembangan perangkat lunak. Memahami dan menerapkan konsep-konsep ini membantu mengatasi tantangan dan mencapai keberhasilan dalam proyek perangkat lunak.

People

Dalam konteks Manajemen Perangkat Lunak, elemen "*People*" (orang-orang) merujuk pada semua individu yang terlibat dalam pengembangan, pengujian, peluncuran, dan pemeliharaan perangkat lunak. Elemen ini sangat penting karena keberhasilan proyek perangkat lunak sangat bergantung pada kemampuan, komunikasi, dan kerjasama dari individu-individu yang terlibat. Berikut adalah beberapa aspek terkait "*People*" dalam manajemen

perangkat lunak:

1. Tim Pengembangan

Ini adalah kelompok orang yang secara langsung terlibat dalam merancang, mengembangkan, dan menguji perangkat lunak. Tim ini biasanya terdiri dari pengembang perangkat lunak, perancang antarmuka pengguna (UI/UX), insinyur kualitas, dan anggota lainnya yang memiliki peran khusus dalam pembuatan produk.

2. Pemangku Kepentingan (Stakeholders)

Pemangku kepentingan adalah individu atau kelompok yang memiliki minat dalam proyek perangkat lunak, termasuk klien, pengguna akhir, manajer, pemilik produk, dan lainnya. Berkomunikasi dengan pemangku kepentingan adalah penting untuk memahami kebutuhan dan harapan mereka.

3. Manajemen Proyek

Manajer proyek bertanggung jawab untuk mengawasi dan mengkoordinasikan aktivitas tim pengembangan. Mereka mengelola perencanaan, alokasi sumber daya, pemantauan kemajuan, dan pelaporan kepada pemangku kepentingan.

4. Keterlibatan Pengguna

Melibatkan pengguna akhir dalam tahap pengembangan, melalui umpan balik dan pengujian beta, membantu memastikan bahwa perangkat lunak memenuhi kebutuhan mereka dan memiliki antarmuka yang mudah digunakan.

5. Keterampilan dan Kepemimpinan

Setiap anggota tim harus memiliki keterampilan yang relevan untuk perannya dalam proyek. Kepemimpinan dalam tim juga penting untuk menginspirasi dan memotivasi anggota tim.

6. Kerjasama dan Kolaborasi

Kolaborasi yang baik di antara anggota tim dan pemangku kepentingan membantu dalam pertukaran ide, pemecahan masalah, dan pengambilan keputusan yang lebih baik.

7. Komunikasi Efektif

Komunikasi yang jelas dan terbuka antara semua pihak terlibat adalah kunci. Ini meliputi berbagi informasi proyek, mengatasi masalah, dan memastikan semua orang memiliki pemahaman yang sama tentang tujuan dan persyaratan.

8. Pengembangan Keterampilan

Tim pengembangan perangkat lunak harus terus meningkatkan keterampilan mereka dalam pengembangan perangkat lunak, teknologi terbaru, serta praktik terbaik dalam industri.

9. Kepatuhan dan Etika

Seluruh tim harus mengikuti praktik etika dan mematuhi standar profesional dalam pengembangan perangkat lunak. Ini mencakup menghormati kode etik, hak kekayaan intelektual, dan privasi pengguna.

10. Diversitas dan Inklusi

Menghargai keberagaman dalam tim dan mengedepankan inklusi membantu dalam mempromosikan perspektif yang beragam dalam pengembangan perangkat lunak, yang dapat menghasilkan solusi yang lebih baik dan inklusif.

Dalam keseluruhan, "People" adalah inti dari manajemen perangkat lunak, karena keterampilan, komunikasi, dan kerjasama yang kuat di antara individu-individu ini adalah yang membentuk fondasi kesuksesan proyek perangkat lunak.

Problem

"*Problem*" (masalah) merujuk pada tantangan, hambatan, atau situasi yang memerlukan pemecahan selama siklus hidup pengembangan perangkat lunak. Mengatasi masalah ini dengan cara yang efektif merupakan bagian penting dari manajemen perangkat lunak yang berhasil. Berikut adalah beberapa aspek terkait "*Problem*" dalam manajemen perangkat lunak:

1. Tantangan Teknis
Proyek pengembangan perangkat lunak seringkali melibatkan tantangan teknis kompleks yang memerlukan pemahaman mendalam tentang teknologi dan solusi kreatif untuk mengatasinya.
2. Keterlambatan Jadwal
Tantangan seperti kurangnya sumber daya, perubahan dalam persyaratan, atau kendala lain dapat menyebabkan keterlambatan dalam jadwal proyek. Mengelola keterlambatan ini dan mengambil tindakan korektif adalah penting.
3. Kualitas Produk
Masalah terkait kualitas, seperti bug atau masalah kinerja, dapat mempengaruhi fungsionalitas dan pengalaman pengguna akhir. Identifikasi, perbaikan, dan pengujian yang cermat diperlukan untuk menjaga kualitas produk.
4. Kesalahpahaman Persyaratan
Jika persyaratan produk tidak dipahami dengan benar oleh tim pengembangan, ini bisa menyebabkan produk yang tidak sesuai dengan ekspektasi pengguna. Komunikasi yang baik dengan pemangku kepentingan membantu mengatasi masalah ini.
5. Keterbatasan Sumber Daya

Terbatasnya anggaran, waktu, atau tenaga kerja bisa menjadi hambatan dalam pengembangan perangkat lunak. Manajemen sumber daya yang cerdas dan pengaturan prioritas diperlukan untuk mengatasi keterbatasan ini.

6. Konflik Tim

Konflik internal dalam tim pengembangan bisa mengganggu kerja sama dan produktivitas. Mengatasi konflik dengan komunikasi terbuka, pengertian, dan pendekatan kolaboratif penting untuk menjaga harmoni dalam tim.

7. Perubahan Persyaratan

Perubahan yang tidak terduga dalam persyaratan proyek dapat mempengaruhi perencanaan dan jalur pengembangan. Manajemen perubahan yang efektif membantu dalam mengatasi dampak negatif dari perubahan ini.

8. Masalah Teknis atau Infrastruktur

Gangguan teknis, kegagalan perangkat keras, atau masalah infrastruktur lainnya dapat mengganggu pengembangan dan pengujian perangkat lunak.

9. Ketidakpastian

Tantangan yang berkaitan dengan ketidakpastian eksternal, seperti perubahan tren pasar atau regulasi, bisa mempengaruhi arah proyek. Fleksibilitas dan adaptabilitas dalam merespons perubahan sangat penting.

10. Kompleksitas Proyek

Beberapa proyek perangkat lunak dapat menjadi sangat kompleks dengan banyaknya fitur, integrasi, atau persyaratan. Mengelola kompleksitas ini melibatkan perencanaan yang baik dan koordinasi yang efektif.

Dalam manajemen perangkat lunak, penting untuk menghadapi masalah dengan kepala dingin, berkolaborasi dengan tim, dan menerapkan solusi yang efektif. Masalah yang diatasi dengan baik dapat menghasilkan produk perangkat lunak yang lebih baik dan mengurangi risiko proyek secara keseluruhan.

Ruang Lingkup Masalah (*Problem Scope*)

Batasan dan cakupan dari masalah yang perlu diatasi dalam proyek pengembangan perangkat lunak. Menetapkan ruang lingkup masalah yang jelas penting untuk memahami batas-batas proyek, menghindari pengembangan yang tidak terkendali, dan memfokuskan upaya tim pada masalah yang paling krusial. Berikut adalah beberapa aspek yang terkait dengan ruang lingkup masalah:

1. Definisi Persyaratan

Menetapkan persyaratan perangkat lunak dengan jelas adalah bagian penting dari menentukan ruang lingkup masalah. Ini mencakup fungsi-fungsi utama, fitur-fitur, dan karakteristik yang diharapkan dari produk perangkat lunak.

2. Batasan Fungsional

Menentukan apa yang masuk dan apa yang tidak masuk ke dalam cakupan perangkat lunak adalah penting. Ini membantu menghindari ambiguitas dan memastikan bahwa perangkat lunak tidak menjadi terlalu kompleks atau tidak terkendali.

3. Batasan Teknis

Mengidentifikasi batasan teknis, seperti platform yang akan digunakan, dukungan perangkat keras, atau kemampuan integrasi, membantu dalam merencanakan pengembangan yang realistis.

4. Batasan Waktu

Menetapkan batas waktu untuk pengembangan dan peluncuran perangkat lunak membantu dalam mengatur ekspektasi dan menghindari keterlambatan yang tidak terduga.

5. Batasan Sumber Daya

Mengidentifikasi sumber daya yang tersedia, termasuk tenaga kerja, anggaran, dan peralatan, membantu dalam mengukur ketersediaan untuk mengatasi masalah.

6. Lingkungan dan Konteks

Memahami konteks operasional perangkat lunak dan lingkungan di mana perangkat lunak akan digunakan membantu dalam merancang solusi yang relevan.

7. Tingkat Prioritas

Mengidentifikasi fitur atau masalah yang memiliki prioritas tinggi membantu dalam pengaturan prioritas pengembangan dan pemecahan masalah.

8. Ekspektasi Pengguna

Memahami ekspektasi pengguna dan kebutuhan mereka membantu dalam menentukan ruang lingkup masalah yang relevan dan bermanfaat bagi mereka.

9. Pemahaman Pemangku Kepentingan

Berkomunikasi dengan pemangku kepentingan (stakeholders) dan memastikan pemahaman yang jelas tentang tujuan dan batasan proyek membantu dalam menetapkan ruang lingkup yang konsisten dengan ekspektasi mereka.

Menetapkan ruang lingkup masalah yang tepat memungkinkan tim pengembangan untuk fokus pada tugas yang paling penting, menghindari proyek yang melampaui batasan yang layak, dan menghasilkan perangkat lunak

yang sesuai dengan kebutuhan dan harapan.

Dekomposisi Masalah (*Problem Decomposition*)

Dekomposisi masalah (problem decomposition) adalah proses memecah masalah yang kompleks menjadi bagian-bagian yang lebih kecil, lebih terkelola, dan lebih mudah dimengerti. Ini adalah teknik penting dalam pengembangan perangkat lunak yang kompleks untuk mengatasi kompleksitas dan memfasilitasi pengembangan yang efisien. Berikut adalah cara-cara di mana dekomposisi masalah digunakan dalam manajemen perangkat lunak:

1. **Pemisahan Fungsionalitas**

Memisahkan fungsionalitas utama menjadi modul atau komponen yang lebih kecil. Setiap modul dapat memiliki tugas dan tanggung jawab tertentu, sehingga memudahkan pengembangan, pengujian, dan pemeliharaan.

2. **Pembagian Tugas Tim**

Pada tingkat tim, masalah besar dapat dibagi menjadi tugas-tugas yang lebih kecil dan lebih terkelola. Ini memungkinkan anggota tim untuk fokus pada tugas-tugas spesifik dan meningkatkan produktivitas.

3. **Dekomposisi Hierarkis**

Masalah kompleks dapat dibagi menjadi tingkatan hierarki, dengan masalah tingkat atas dibagi menjadi sub-masalah yang lebih kecil. Setiap sub-masalah kemudian dapat dipecah lagi menjadi sub-sub-masalah yang lebih kecil lagi.

4. **Pemecahan dalam Aliran Kerja**

Proses kerja atau alur kerja dalam perangkat lunak dapat dipecah menjadi langkah-langkah yang lebih kecil danurut. Ini membantu dalam memahami

urutan tindakan dan mengidentifikasi area yang memerlukan perhatian khusus.

5. Pemisahan dalam Tampilan dan Kontrol

Dalam pengembangan antarmuka pengguna (UI/UX), dekomposisi dapat melibatkan pemisahan tampilan (UI) dari logika kontrol (backend). Ini memisahkan perhatian antara penampilan yang disajikan kepada pengguna dan proses di belakang layar.

6. Dekomposisi dalam Kasus Pengguna

Memisahkan perangkat lunak ke dalam kasus pengguna (use cases) atau skenario-skenario dapat membantu dalam merancang solusi yang lebih fokus pada kebutuhan pengguna.

7. Pemecahan dalam Modul Fungsional

Jika perangkat lunak melibatkan berbagai modul atau komponen yang berfungsi secara terpisah, dekomposisi dapat membantu dalam merancang dan mengembangkan setiap modul dengan lebih baik.

8. Pemisahan untuk Pengujian

Dekomposisi masalah juga dapat membantu dalam merencanakan pengujian yang lebih efektif. Dengan memecah masalah menjadi bagian-bagian yang lebih kecil, pengujian dapat difokuskan pada setiap bagian secara terpisah.

Dekomposisi masalah membantu dalam mengatasi kompleksitas dan memungkinkan tim untuk bekerja pada bagian-bagian perangkat lunak secara lebih fokus. Hal ini juga mendukung pengembangan paralel, karena bagian-bagian yang berbeda dapat dikerjakan secara independen oleh anggota tim yang berbeda.

Proses

Istilah "Proses" mengacu pada langkah-langkah yang diambil untuk mengarahkan, mengelola, dan mengendalikan berbagai aspek yang terlibat dalam pengembangan, pengujian, dan pemeliharaan perangkat lunak. Proses dalam manajemen perangkat lunak membentuk kerangka kerja untuk menjalankan kegiatan-kegiatan yang diperlukan untuk mencapai tujuan proyek secara efisien dan efektif. Berikut adalah beberapa konsep terkait "Proses" dalam manajemen perangkat lunak:

1. **Proses Pengembangan**
Ini adalah langkah-langkah yang diambil dari awal hingga akhir dalam menciptakan perangkat lunak. Ini termasuk perencanaan, analisis, desain, implementasi, pengujian, dan peluncuran.
2. **Proses Pengujian**
Langkah-langkah yang dilakukan untuk memverifikasi dan memvalidasi fungsionalitas, kualitas, dan kinerja perangkat lunak. Ini melibatkan pengujian fungsional, pengujian integrasi, pengujian kinerja, dan lainnya.
3. **Proses Pemeliharaan**
Setelah perangkat lunak diluncurkan, proses pemeliharaan melibatkan pemantauan kinerja, pembaruan perangkat lunak, penanganan masalah yang muncul, dan pengembangan fitur baru.
4. **Proses Manajemen Proyek**
Ini melibatkan perencanaan, pengorganisasian, pelaksanaan, pengendalian, dan pemantauan aktivitas proyek. Tujuannya adalah mengelola sumber daya dan risiko dengan cara yang efisien.
5. **Proses Pengendalian Versi**
Mengelola perubahan dan evolusi perangkat lunak

dengan menggunakan sistem pengendalian versi. Ini memastikan bahwa semua perubahan direkam, dikelola, dan disinkronkan di seluruh tim.

6. **Proses Manajemen Risiko**
Identifikasi, analisis, penanganan, dan pengawasan risiko yang mungkin timbul selama siklus hidup proyek. Ini membantu dalam mengantisipasi dan mengurangi dampak dari risiko.
7. **Proses Manajemen Konfigurasi**
Mengelola komponen perangkat lunak yang berubah selama waktu. Ini melibatkan pelacakan perubahan, mengelola versi, dan memastikan konsistensi dalam pengembangan.
8. **Proses Manajemen Perubahan**
Mengelola perubahan dalam persyaratan atau tujuan proyek. Ini melibatkan penilaian, persetujuan, dan implementasi perubahan yang diusulkan.
9. **Proses Pengiriman Produk**
Persiapan dan pengiriman produk perangkat lunak yang sudah siap kepada pengguna akhir atau klien. Ini termasuk pengujian akhir, pelatihan, dan penyerahan produk.
10. **Proses Komunikasi**
Ini mencakup komunikasi antara anggota tim, pemangku kepentingan, dan pemakai akhir. Komunikasi yang jelas dan terbuka penting untuk kesuksesan proyek.
11. **Proses Evaluasi dan Pemantauan**
Pemantauan terus-menerus terhadap kemajuan proyek, pengukuran kinerja, dan evaluasi hasil untuk memastikan bahwa proyek berjalan sesuai rencana.
12. **Proses Peningkatan**
Melibatkan refleksi dan pembelajaran dari proyek sebelumnya, serta menerapkan pengetahuan baru ke

dalam proses pengembangan perangkat lunak selanjutnya.

Setiap proses memiliki langkah-langkah dan praktik-praktik yang terkait untuk memastikan keberhasilan proyek perangkat lunak. Mengelola proses dengan baik membantu dalam mengatasi kompleksitas dan risiko yang terkait dengan pengembangan perangkat lunak.

Menggabungkan Masalah dan Proses

Dalam pengembangan perangkat lunak, masalah adalah hambatan atau tantangan yang perlu diatasi, sementara proses adalah kerangka kerja yang digunakan untuk mengarahkan langkah-langkah dalam mencapai tujuan proyek. Berikut adalah bagaimana kedua konsep ini dapat saling berinteraksi:

1. Identifikasi Masalah

Pada awal proyek, tim perlu mengidentifikasi masalah atau tantangan yang perlu diatasi. Ini bisa berupa persyaratan yang kompleks, hambatan teknis, atau tantangan lainnya yang mungkin muncul selama pengembangan.

2. Pemetaan Proses

Setelah masalah diidentifikasi, tim perlu memetakan proses yang akan digunakan untuk mengatasi masalah tersebut. Ini melibatkan menentukan langkah-langkah yang diperlukan, sumber daya yang akan digunakan, dan tahapan pengembangan yang relevan.

3. Dekomposisi Masalah

Jika masalah yang dihadapi terlalu kompleks, tim dapat menerapkan dekomposisi masalah untuk memecahnya menjadi bagian-bagian yang lebih kecil dan lebih terkelola. Ini membantu dalam merancang

proses pengembangan yang lebih fokus.

4. Perencanaan Proses

Proses pengembangan perangkat lunak harus direncanakan dengan cermat berdasarkan masalah yang dihadapi. Ini melibatkan menetapkan tahapan pengembangan, alokasi sumber daya, jadwal, dan tanggung jawab tim.

5. Pelaksanaan Proses

Tim melaksanakan proses pengembangan sesuai rencana yang telah dibuat. Langkah-langkah ini termasuk merancang, mengembangkan, menguji, dan mengimplementasikan solusi untuk mengatasi masalah yang diidentifikasi.

6. Pemantauan dan Penyesuaian

Selama pelaksanaan proses, tim memantau kemajuan proyek dan mengukur apakah solusi yang diimplementasikan berhasil mengatasi masalah yang dihadapi. Jika perlu, penyesuaian dapat dilakukan untuk memastikan tujuan tetap tercapai.

7. Manajemen Risiko

Proses ini melibatkan identifikasi dan pengelolaan risiko yang mungkin timbul selama pengembangan. Jika masalah muncul selama pelaksanaan, tim dapat merencanakan tindakan mitigasi yang sesuai.

8. Kontinuitas Pemeliharaan

Setelah peluncuran, proses pemeliharaan memastikan bahwa solusi tetap berfungsi dengan baik dan masalah yang mungkin muncul diatasi dengan cepat.

9. Evaluasi dan Peningkatan

Setelah selesai, proses harus dievaluasi untuk mengidentifikasi apa yang berhasil dan apa yang dapat ditingkatkan. Pembelajaran dari proses ini dapat diterapkan pada proyek-proyek selanjutnya.

Menggabungkan konsep "Masalah" dan "Proses" membantu tim dalam merencanakan, melaksanakan, dan mengelola proyek perangkat lunak dengan cara yang sistematis dan terorganisir. Ini juga membantu dalam mengatasi tantangan dan menghasilkan solusi yang berkualitas untuk masalah yang dihadapi.

Dekomposisi Proses

Dekomposisi proses adalah pendekatan untuk memecah suatu proses yang kompleks menjadi langkah-langkah yang lebih kecil dan terkelola. Ini membantu dalam mengatasi kompleksitas, memfasilitasi pemahaman yang lebih baik, dan mengelola setiap langkah dengan lebih efisien. Dalam konteks Manajemen Perangkat Lunak, dekomposisi proses dapat diterapkan untuk berbagai tahap pengembangan, pengujian, atau pemeliharaan perangkat lunak. Berikut adalah langkah-langkah dalam dekomposisi proses:

1. **Identifikasi Tahapan Utama**
Identifikasi tahapan-tahapan besar dalam proses yang ingin Anda dekomposisi. Misalnya, dalam proses pengembangan perangkat lunak, tahapan-tahapan tersebut mungkin termasuk perencanaan, analisis, desain, implementasi, pengujian, dan peluncuran.
2. **Pemecahan Menjadi Sub-Tahapan**
Untuk setiap tahapan utama, pecah menjadi sub-tahapan yang lebih kecil. Misalnya, dalam tahap analisis, Anda bisa memecahnya menjadi sub-tahapan seperti pengumpulan persyaratan, analisis kebutuhan, dan perancangan konsep.
3. **Definisikan Tugas-tugas Spesifik**
Dalam setiap sub-tahapan, identifikasi tugas-tugas

spesifik yang perlu dilakukan. Misalnya, dalam sub-tahapan pengumpulan persyaratan, tugas-tugasnya mungkin termasuk wawancara pengguna, pengumpulan dokumen, dan identifikasi persyaratan.

4. Alokasi Sumber Daya

Tentukan sumber daya yang diperlukan untuk menyelesaikan setiap tugas. Ini mencakup tenaga kerja, waktu, alat, dan bahan-bahan lain yang dibutuhkan.

5. Rangkaian Prioritas

Prioritaskan tugas-tugas berdasarkan urgensi dan dampak pada hasil akhir. Ini membantu dalam mengatur urutan penyelesaian tugas.

6. Penugasan dan Tanggung Jawab

Setelah tugas-tugas ditetapkan, tentukan siapa yang bertanggung jawab atas setiap tugas. Ini memastikan klarifikasi peran dan tanggung jawab di antara anggota tim.

7. Pemantauan dan Pelaporan

Setelah tugas-tugas dimulai, pantau kemajuan dan laporkan perkembangan kepada pemangku kepentingan. Ini membantu dalam memastikan penyelesaian tepat waktu dan mengatasi masalah jika muncul.

8. Integrasi dan Koordinasi

Pastikan bahwa tugas-tugas yang terpisah ini tetap terintegrasi dengan baik. Koordinasi yang efektif antara tim adalah kunci kesuksesan dalam dekomposisi proses.

9. Evaluasi dan Peningkatan

Setelah proses selesai, evaluasi efektivitas dekomposisi proses. Apakah memudahkan pelaksanaan? Apakah ada bagian yang

membutuhkan peningkatan?

Dekomposisi proses membantu dalam mengelola proyek yang kompleks dengan lebih terstruktur. Ini memungkinkan tim untuk fokus pada tugas-tugas spesifik, menghindari kebingungan, dan mengurangi risiko kesalahan.

BAB 2 KONSEP PERANGKAT LUNAK

Pendahuluan

Bab ini akan membahas konsep-konsep dasar yang terkait dengan perangkat lunak. Perangkat lunak adalah komponen inti dalam sistem komputer yang memungkinkan pengguna untuk melakukan berbagai tugas dan memanfaatkan keunggulan teknologi. Pemahaman tentang konsep perangkat lunak sangat penting dalam pengembangan, penggunaan, dan pemeliharaan sistem yang efektif dan efisien. Perangkat lunak merupakan kumpulan instruksi atau program komputer yang dirancang untuk menjalankan tugas tertentu. Ini mencakup aplikasi desktop, aplikasi seluler, sistem operasi, perangkat lunak perangkat keras tertanam, dan banyak lagi. Perangkat lunak memiliki peran penting dalam dunia teknologi dan masyarakat modern, memfasilitasi berbagai kegiatan dan meningkatkan efisiensi serta produktivitas.

Definisi Perangkat Lunak

Perangkat lunak (*software*) adalah kumpulan instruksi komputer, program, dan data yang memberikan instruksi kepada komputer untuk menjalankan tugas tertentu. Perangkat lunak tidak terlihat secara fisik, melainkan merupakan komponen abstrak yang berjalan di atas perangkat keras komputer. Definisi umum dari perangkat lunak adalah sebagai serangkaian program komputer, data, dan instruksi yang mengontrol

operasi perangkat keras komputer dan memberikan fungsi-fungsi yang diperlukan untuk menjalankan tugas-tugas khusus. Perangkat lunak mencakup sistem operasi, aplikasi, utilitas, perangkat lunak pengembangan, dan lainnya.

Perangkat lunak dapat dibagi menjadi dua kategori utama:

1. Sistem Operasi (*Operating System*)

Merupakan perangkat lunak dasar yang mengelola sumber daya perangkat keras komputer, seperti memori, prosesor, perangkat input/output, dan memberikan layanan dasar kepada aplikasi lainnya. Contoh sistem operasi termasuk Windows, macOS, Linux, dan Android.

2. Aplikasi (*Application*)

Merupakan perangkat lunak yang dirancang untuk menjalankan tugas-tugas spesifik dan memenuhi kebutuhan pengguna. Aplikasi dapat mencakup berbagai jenis seperti aplikasi produktivitas (seperti pengolah kata dan spreadsheet), aplikasi multimedia (seperti pemutar musik dan pemutar video), aplikasi permainan (*game*), aplikasi web, dan banyak lagi.

Perangkat lunak dapat dikembangkan melalui proses pengkodean dan pemrograman oleh para pengembang perangkat lunak. Mereka menggunakan berbagai bahasa pemrograman dan alat pengembangan untuk membuat perangkat lunak yang fungsional, efisien, dan mudah digunakan.

Jenis-jenis Perangkat Lunak

Pada bagian ini, kita akan mempelajari berbagai jenis perangkat lunak yang ada. Klasifikasi perangkat lunak dapat dibagi menjadi beberapa kategori, seperti perangkat

lunak sistem, perangkat lunak aplikasi, dan perangkat lunak tengah (middleware). Penjelasan tentang masing-masing jenis perangkat lunak akan diberikan, bersama dengan contoh-contohnya.



Gambar 1. Jenis perangkat lunak (www.gramedia.com)

Perangkat Lunak Sistem

Perangkat lunak sistem adalah perangkat lunak yang bertanggung jawab untuk mengelola dan mengendalikan operasi perangkat keras serta menyediakan layanan dasar bagi perangkat lunak lainnya. Beberapa contoh perangkat lunak sistem termasuk:

1. **Sistem Operasi (*Operating System*)**
Merupakan perangkat lunak inti yang mengelola sumber daya komputer, seperti memori, pemrosesan, pengaturan jaringan, dan interaksi dengan perangkat keras. Contoh sistem operasi termasuk Windows, macOS, dan Linux.
2. **Driver Perangkat Keras (*Device Drivers*)**
Perangkat lunak ini digunakan untuk menghubungkan perangkat keras dengan sistem operasi. Driver perangkat keras memungkinkan sistem operasi mengenali dan berinteraksi dengan

perangkat keras, seperti printer, kartu grafis, atau mouse.

3. *Utilitas Sistem (System Utilities)*

Merupakan perangkat lunak yang membantu mengelola dan memelihara sistem operasi, seperti utilitas antivirus, utilitas pemulihan data, atau alat konfigurasi jaringan.

Perangkat Lunak Aplikasi

Perangkat lunak aplikasi adalah perangkat lunak yang dirancang untuk memenuhi kebutuhan khusus pengguna atau organisasi. Jenis perangkat lunak ini meliputi:

1. *Pengolah Kata (Word Processors)*

Perangkat lunak ini digunakan untuk membuat, mengedit, dan memformat dokumen teks, seperti Microsoft Word atau Google Docs.

2. *Spreadsheet (Spreadsheet Software)*

Perangkat lunak ini memungkinkan pengguna untuk membuat, mengorganisir, dan menganalisis data dalam bentuk tabel dan rumus matematika, seperti Microsoft Excel atau Google Sheets.

3. *Pengelola Proyek (Project Management Software)*

Digunakan untuk merencanakan, melacak, dan mengelola proyek, termasuk pengaturan jadwal, tugas, sumber daya, dan kolaborasi tim. Contohnya adalah Microsoft Project atau Trello.

4. *Sistem Manajemen Basis Data (Database Management System - DBMS)*

Merupakan perangkat lunak untuk membuat, mengelola, dan mengakses basis data. Contoh DBMS meliputi MySQL, Oracle, dan Microsoft SQL Server.

5. *Program Grafis (Graphic Design Software)*

Perangkat lunak ini memungkinkan pengguna untuk membuat dan mengedit gambar, grafik, atau desain visual, seperti Adobe Photoshop atau CorelDRAW.

Perangkat Lunak Tengah (Middleware)

Perangkat Lunak Tengah, yang juga dikenal sebagai Middleware, berperan sebagai perantara antara perangkat lunak aplikasi dan perangkat keras. Middleware menyediakan lingkungan yang terpusat untuk berbagai aplikasi berkomunikasi, berintegrasi, dan beroperasi secara efisien. Berikut adalah beberapa jenis perangkat lunak tengah yang umum digunakan:

1. Enterprise Service Bus (ESB)
ESB adalah platform middleware yang memfasilitasi integrasi aplikasi dan layanan dalam lingkungan enterprise. ESB menyediakan kerangka kerja untuk menghubungkan, mengarahkan, dan mengelola aliran data antara berbagai sistem dan komponen aplikasi yang berbeda.
2. Message Queue (MQ) Middleware
Middleware antrian pesan digunakan untuk mengirim dan menerima pesan antara komponen aplikasi yang terpisah secara asinkron. Ini membantu dalam pemrosesan pesan yang dapat diandalkan dan berorientasi waktu.
3. Sistem Manajemen Basis Data (*Database Management System - DBMS*)
DBMS adalah perangkat lunak yang memungkinkan pembuatan, pengelolaan, dan pengaksesan basis data. Middleware basis data membantu dalam menghubungkan aplikasi dengan basis data yang berbeda dan menyediakan antarmuka standar untuk berinteraksi dengan basis data.

4. Sistem Manajemen Konten (*Content Management System - CMS*)

CMS adalah perangkat lunak yang digunakan untuk membuat, mengelola, dan mengatur konten digital seperti situs web, blog, atau portal. Middleware CMS menyediakan fitur seperti manajemen otorisasi, manajemen aliran kerja, dan pengaturan tampilan konten yang mempermudah pengelolaan konten.

5. Sistem Manajemen Layanan (*Service Management System*)

Middleware manajemen layanan membantu dalam manajemen dan pengaturan layanan IT dalam organisasi. Ini termasuk manajemen katalog layanan, manajemen permintaan, manajemen pemantauan, dan manajemen perubahan.

6. Sistem Manajemen Transaksi (*Transaction Management System*)

Middleware manajemen transaksi digunakan untuk mengelola transaksi dalam lingkungan yang terdistribusi dan transaksional. Ini menyediakan mekanisme untuk memastikan konsistensi dan integritas data dalam transaksi yang melibatkan beberapa sistem.

7. Sistem Manajemen Jaringan (*Network Management System*)

Middleware manajemen jaringan digunakan untuk memantau, mengelola, dan mengoptimalkan kinerja jaringan. Ini meliputi pemantauan jaringan, manajemen perangkat jaringan, manajemen lalu lintas, dan manajemen keamanan jaringan.

8. Middleware Berbasis Layanan (*Service-Oriented Middleware*)

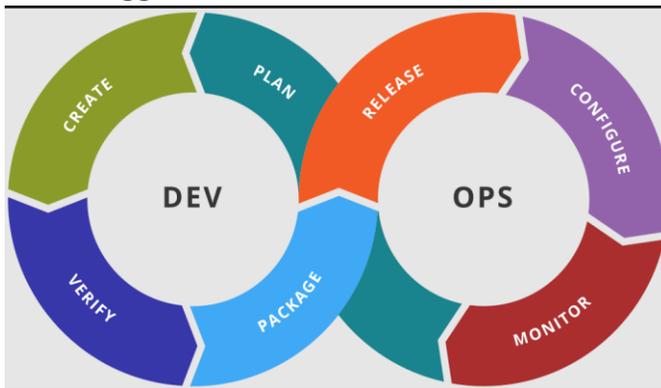
Middleware berbasis layanan digunakan untuk membangun dan mengelola arsitektur berorientasi

layanan (Service-Oriented Architecture - SOA). Ini memfasilitasi integrasi aplikasi yang longgar melalui layanan yang berinteraksi satu sama lain menggunakan protokol standar seperti SOAP (Simple Object Access Protocol) atau REST (Representational State Transfer).

Perangkat lunak tengah ini memainkan peran kunci dalam memfasilitasi integrasi sistem dan komunikasi antara aplikasi yang berbeda.

Proses Pengembangan Perangkat Lunak

Di sini, kita akan menjelajahi tahapan-tahapan dalam pengembangan perangkat lunak. Ini meliputi analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan. Setiap tahapan akan dijelaskan secara rinci dan pentingnya dalam mencapai perangkat lunak yang berkualitas tinggi akan ditekankan.



Gambar 2. Pengembangan Perangkat Lunak
(<http://microdataindonesia.co.id/>)

Tahapan pengembangan perangkat lunak dapat bervariasi tergantung pada metodologi pengembangan yang digunakan, seperti Waterfall, Agile, atau DevOps. Berikut adalah tahapan umum yang sering terjadi dalam

pengembangan perangkat lunak:

1. Analisis Kebutuhan
 - a. Identifikasi kebutuhan pengguna dan pemangku kepentingan.
 - b. Analisis masalah yang ingin dipecahkan oleh perangkat lunak.
 - c. Definisi tujuan dan batasan proyek.
2. Perencanaan
 - a. Penentuan anggaran, jadwal, dan sumber daya yang diperlukan.
 - b. Pembuatan rencana pengembangan yang mencakup tahapan, aktivitas, dan tanggung jawab.
3. Desain
 - a. Perancangan arsitektur sistem.
 - b. Pembuatan desain perangkat lunak, termasuk desain antarmuka pengguna, struktur database, dan algoritma.
 - c. Identifikasi komponen perangkat lunak yang perlu dikembangkan.
4. Implementasi
 - a. Pengembangan kode program berdasarkan desain yang telah dibuat.
 - b. Pengujian komponen perangkat lunak secara terpisah (unit testing).
 - c. Integrasi komponen perangkat lunak dan pengujian integrasi.
5. Pengujian dan Verifikasi
 - a. Pengujian keseluruhan sistem perangkat lunak untuk memastikan kinerja, keandalan, dan keamanan.
 - b. Identifikasi dan perbaikan bug atau masalah yang ditemukan.
 - c. Verifikasi apakah perangkat lunak memenuhi

persyaratan dan kebutuhan yang telah ditetapkan.

6. Penyampaian dan Implementasi
 - a. Pindahan perangkat lunak ke lingkungan produksi atau pengguna akhir.
 - b. Instalasi dan konfigurasi perangkat lunak di lingkungan yang relevan.
 - c. Pelatihan pengguna dan pemangku kepentingan terkait.
7. Pemeliharaan
 - a. Pemantauan dan pemecahan masalah setelah perangkat lunak digunakan.
 - b. Peningkatan perangkat lunak berdasarkan umpan balik pengguna dan perubahan kebutuhan.
 - c. Pemeliharaan rutin, pembaruan, dan perbaikan bug yang terjadi.

Tahapan-tahapan ini dapat dilakukan secara berurutan (linier) dalam model Waterfall atau secara iteratif dalam metodologi Agile, dengan siklus pengembangan yang berulang. Setiap tahapan dapat melibatkan aktivitas tambahan dan alat bantu yang sesuai perencanaan.

Metodologi Pengembangan Perangkat

Lunak

Bagian ini akan membahas berbagai metodologi pengembangan perangkat lunak yang umum digunakan, seperti waterfall, agile, scrum, dan spiral. Setiap metodologi akan dijelaskan secara singkat, termasuk kelebihan dan kelemahannya, serta situasi di mana metodologi tersebut lebih cocok digunakan.

Berikut adalah beberapa metodologi pengembangan

perangkat lunak yang umum digunakan:

1. Metodologi Waterfall

- a. Pendekatan pengembangan perangkat lunak linear, dengan tahapan yang dilakukan secara berurutan.
- b. Tahapan utama meliputi analisis kebutuhan, desain, implementasi, pengujian, dan penyampaian.
- c. Setiap tahapan harus selesai sebelum memasuki tahapan berikutnya.
- d. Cocok untuk proyek dengan kebutuhan yang stabil dan jelas, serta dengan tim yang terstruktur dan hierarkis.

2. Metodologi Agile

- a. Pendekatan pengembangan perangkat lunak yang adaptif dan berfokus pada kerja tim yang kolaboratif dan responsif terhadap perubahan.
- b. Mengutamakan pengiriman perangkat lunak dalam iterasi singkat, yang dikenal sebagai sprint.
- c. Komunikasi yang intensif antara anggota tim dan pemangku kepentingan.
- d. Beberapa kerangka kerja Agile yang populer termasuk Scrum, Kanban, dan Extreme Programming (XP).

3. Metodologi Scrum

- a. Salah satu kerangka kerja Agile yang paling umum digunakan.
- b. Memecah pekerjaan menjadi iterasi waktu yang terbatas, disebut sprint, biasanya dalam rentang waktu 1-4 minggu.
- c. Anggota tim bekerja secara kolaboratif dalam tim kecil yang terorganisir dalam peran

- seperti Scrum Master, Product Owner, dan Tim Pengembang.
- d. Dilakukan pertemuan reguler, seperti Daily Stand-up, Sprint Planning, Sprint Review, dan Sprint Retrospective.
4. Metodologi Spiral
- a. Pendekatan pengembangan perangkat lunak yang menggabungkan elemen dari model waterfall dengan siklus iteratif.
 - b. Proses pengembangan berputar di sekitar tahapan identifikasi risiko, analisis dan perencanaan, pembuatan prototipe, dan evaluasi.
 - c. Berfokus pada pengelolaan risiko dan adaptasi berdasarkan umpan balik dari setiap siklus.

Metodologi pengembangan perangkat lunak yang dipilih tergantung pada kebutuhan proyek, kompleksitasnya, lingkungan kerja tim, dan tingkat ketidakpastian yang ada. Adopsi metodologi yang tepat dapat membantu tim pengembangan dalam mengelola proyek dengan lebih efektif dan menghasilkan perangkat lunak yang berkualitas.

Manajemen Konfigurasi Perangkat Lunak

Manajemen konfigurasi perangkat lunak sangat penting dalam pengembangan perangkat lunak yang kompleks. Bagian ini akan membahas konsep manajemen konfigurasi, termasuk kontrol versi, manajemen perubahan, dan pengelolaan dependensi antar komponen perangkat lunak.

Manajemen konfigurasi perangkat lunak adalah praktik untuk mengelola perubahan dan pengendalian versi

dalam pengembangan perangkat lunak. Ini melibatkan pengaturan, pelacakan, dan pengelolaan semua elemen konfigurasi perangkat lunak, termasuk kode sumber, dokumen, data, dan komponen terkait lainnya. Konsep manajemen konfigurasi meliputi kontrol versi, manajemen perubahan, dan pengelolaan dependensi antar komponen perangkat lunak.

1. Kontrol Versi

- a. Kontrol versi melibatkan pelacakan dan pengelolaan setiap perubahan yang terjadi pada kode sumber dan komponen perangkat lunak lainnya.
- b. Menggunakan sistem kontrol versi (version control system) untuk mengelola revisi kode sumber dan dokumen, serta memungkinkan kolaborasi tim.
- c. Memungkinkan tim pengembang untuk bekerja secara paralel, melacak perubahan, dan mengembalikan ke versi sebelumnya jika diperlukan.
- d. Contoh alat kontrol versi yang umum digunakan adalah Git, Subversion (SVN), dan Mercurial.

2. Manajemen Perubahan

- a. Manajemen perubahan melibatkan pengelolaan dan pengendalian perubahan yang terjadi selama siklus pengembangan perangkat lunak.
- b. Memiliki proses yang terstruktur untuk mengelola permintaan perubahan, mengevaluasi dampaknya, dan mengimplementasikan perubahan dengan minimal gangguan.
- c. Melibatkan penerapan prosedur perubahan,

- pengujian perubahan, dan pelacakan status perubahan hingga selesai.
- d. Berguna untuk meminimalkan risiko perubahan yang dapat mempengaruhi stabilitas dan kualitas perangkat lunak.
3. Pengelolaan Dependensi Antar Komponen Perangkat Lunak
- a. Perangkat lunak sering terdiri dari berbagai komponen yang saling bergantung satu sama lain.
 - b. Pengelolaan dependensi melibatkan identifikasi, pemantauan, dan manajemen dependensi antara komponen perangkat lunak.
 - c. Menggunakan alat dan teknik untuk memastikan bahwa semua komponen yang diperlukan tersedia dan kompatibel.
 - d. Penting untuk mengelola dependensi dengan baik agar perangkat lunak dapat berfungsi dengan baik dan terhindar dari konflik atau kegagalan saat menjalankan atau mengintegrasikan komponen.

Dalam pengembangan perangkat lunak yang kompleks, manajemen konfigurasi menjadi penting untuk memastikan konsistensi, pengendalian, dan koordinasi antara tim pengembang. Praktik ini membantu dalam melacak perubahan, mengurangi risiko konflik dan kegagalan, serta memungkinkan pengembang untuk bekerja secara kolaboratif dalam menghasilkan perangkat lunak yang stabil dan berkualitas.

Lisensi Perangkat Lunak

Pada bagian terakhir ini, kita akan mempelajari

tentang lisensi perangkat lunak. Kita akan memahami jenis-jenis lisensi perangkat lunak yang ada, seperti lisensi sumber terbuka (open source) dan lisensi properti (proprietary). Implikasi hukum dan etika penggunaan perangkat lunak juga akan dibahas.

Ada beberapa jenis lisensi perangkat lunak yang umum digunakan, yang dapat dibagi menjadi dua kategori utama: lisensi sumber terbuka (open source) dan lisensi properti (proprietary).

1. Lisensi Sumber Terbuka (Open Source):

- a. Lisensi MIT

Lisensi yang memungkinkan pengguna untuk menggunakan, memodifikasi, dan mendistribusikan perangkat lunak dengan sedikit batasan. Lisensi ini mempromosikan kebebasan pengguna dan pengembang.

- b. Lisensi GNU General Public License (GPL)

Lisensi yang mewajibkan perangkat lunak yang diturunkan dari perangkat lunak berlisensi GPL untuk tetap bersifat open source. Jika ada perubahan atau distribusi, kode sumber harus tetap tersedia.

- c. Lisensi Apache

Lisensi yang memberikan kebebasan pengguna untuk menggunakan, mengubah, dan mendistribusikan perangkat lunak. Lisensi ini juga melindungi hak cipta, paten, dan merek dagang pengembang.

- d. Lisensi Creative Commons

Lisensi yang digunakan untuk konten kreatif seperti gambar, musik, atau tulisan. Lisensi ini memberikan pilihan fleksibilitas dalam penggunaan, penggabungan, dan redistribusi konten.

2. Lisensi Properti (Proprietary)

a. Lisensi Perangkat Lunak Komersial

Lisensi yang diberikan oleh perusahaan atau individu yang memiliki hak cipta perangkat lunak. Pengguna harus membeli atau memperoleh izin untuk menggunakan, memodifikasi, atau mendistribusikan perangkat lunak.

b. Lisensi Perangkat Lunak Tertutup

Lisensi yang memberikan hak eksklusif kepada pemilik perangkat lunak. Pengguna diberikan hak terbatas untuk menggunakan perangkat lunak tanpa hak untuk melihat, memodifikasi, atau mendistribusikan kode sumber.

c. Lisensi Perangkat Lunak Freeware

Lisensi yang memungkinkan pengguna untuk menggunakan perangkat lunak tanpa biaya. Namun, hak penggunaan dan distribusi dapat memiliki batasan tertentu.

d. Lisensi Perangkat Lunak Shareware

Lisensi yang memungkinkan pengguna untuk mencoba perangkat lunak secara gratis dalam jangka waktu tertentu, tetapi memerlukan pembayaran jika ingin melanjutkan penggunaan setelah periode tersebut.

Setiap lisensi memiliki peraturan dan ketentuan yang berbeda, dan penting bagi pengguna perangkat lunak untuk memahami lisensi yang digunakan dalam rangka mematuhi aturan dan hak cipta yang berlaku.

BAB 3 KERANGKA KERJA MANAJEMEN PROYEK PERANGKAT LUNAK

Pendahuluan

Banyak Organisasi dan perorangan saat ini memiliki ketertarikan dalam melakukan manajemen proyek. Pada awal abad 20-an hingga tahun 1980-an, manajemen proyek berfokus pada penjadwalan dan sumber daya di bidang industry, militer, komputer, dan konstruksi (Schwalbe, 2016). Saat ini, manajemen proyek melibatkan banyak hal, terutama pada dunia industri ditiap negara. Teknologi baru telah menjadi sangat penting bagi banyak perusahaan. Penggunaan jaringan, perangkat lunak, perangkat keras, dan tim kerja yang tersebar di seluruh dunia telah mengubah lingkungan kerja secara dramatis.

Persaingan global saat ini menunjukkan betapa pentingnya manajemen proyek dalam masyarakat saat ini, terutama dalam proyek-proyek yang melibatkan perangkat lunak. Kompleksitas dan pentingnya proyek perangkat lunak, yang melibatkan penggunaan perangkat keras, aplikasi pendukung dan jaringan dalam menciptakan produk dan layanan telah berkembang sangat pesat. Perusahaan swasta, pemerintahan, dan organisasi nirlaba saat ini menyadari bahwa untuk menjadi sukses, perlu menggunakan konsep manajemen proyek modern, terutama untuk proyek-proyek Teknologi Informasi dan Komunikasi (TIK). Seseorang menyadari bahwa untuk tetap kompetitif di tempat kerja, maka harus mengembangkan keterampilan untuk menjadi anggota tim

proyek dan manajer proyek yang baik. Konsep manajemen proyek akan membantu dan memfasilitasi dalam kehidupan sehari-hari untuk beraktivitas dengan sesamanya dan teknologi.

Manajemen proyek perangkat lunak memainkan peran yang tak tergantikan dalam melaksanakan tugasnya di setiap organisasi pemerintahan, swasta, publik, dan nirlaba. Hal ini memungkinkan untuk melakukan berbagai kegiatan dalam rentang dan kualitas, dalam jangka waktu yang diperlukan dan tanpa melebihi anggaran atau bahkan dengan hasil yang lebih baik dari yang diharapkan. Selama dua dekade terakhir, perangkat lunak telah mengambil dimensi baru untuk memenuhi kebutuhan dunia nyata (Kundu et al., 2015). Revolusi di bidang TIK sangat mempengaruhi strategi operasional organisasi, di semua tingkatan di seluruh dunia. Organisasi dipaksa untuk mengadopsi TIK dalam meningkatkan kontrol yang efektif atas operasi dan indeks produktivitas. Dengan demikian TIK mempengaruhi organisasi di semua bidang seperti; keuangan, pemasaran, pribadi, kesehatan, dan lain-lain.

Saat ini, pengembangan perangkat lunak yang efisien dan murah sangat penting untuk memenuhi kebutuhan TIK yang terus meningkat. Perangkat lunak diperlukan untuk teknologi tinggi dan sistem yang kompleks, sehingga meningkatkan ukuran program yang dibutuhkan. Seiring bertambahnya ukuran program, jumlah orang yang diperlukan dalam proyek juga meningkat. Hal ini meningkatkan biaya, terutama dalam manajemen personalia dan waktu yang berakibat pada produktivitas kelompok pengembangan secara keseluruhan. Berlandaskan hal tersebut maka diperlukan konsep manajemen proyek perangkat lunak sebagai solusinya, sehingga variable-variable tersebut dapat dikontrol dengan manajemen yang baik.

Manajemen Proyek Perangkat Lunak

Proyek adalah usaha singkat untuk mengembangkan produk dan jasa, dengan hasil yang berbeda (Marchewka, 2003). Proyek yang sukses tidak terjadi secara spontan tetapi melalui sejumlah persiapan dan perencanaan terjadi terlebih dahulu (Jahan et al., 2019). Proyek adalah sebuah upaya yang dilakukan oleh sebuah tim atau organisasi yang disengaja dan direncanakan (Schwalbe, 2016). Berdasarkan beberapa pendapat tersebut dapat disintesis bahwa proyek adalah sebuah upaya untuk mengembangkan produk dan layanan yang dilakukan oleh sebuah tim melalui perencanaan, analisis biaya dan penjadwalan waktu.

Manajemen proyek merupakan implementasi pengetahuan, kemampuan, alat, dan Teknik yang digunakan dalam pelaksanaan proyek (Institute, 2014). Ini dapat dicapai dengan menerapkan dan mengintegrasikan proses manajemen proyek, yang terdiri dari analisis kebutuhan, perencanaan, pelaksanaan, pengawasan, pengendalian, dan pelaporan. Manajemen proyek adalah perencanaan, pengorganisasian, pemantauan, pengendalian dan kepemimpinan yang terlibat untuk mencapai tujuan proyek yang aman dengan disepakati untuk waktu, biaya, ruang lingkup dan kualitas (IPMA, 2015). Keberhasilan manajemen proyek sangat bergantung pada penerapan alat yang sesuai (Paliktzoglou, 2015). Manajemen proyek melibatkan sejumlah alat dan teknik, di mana yang disebutkan dalam standar manajemen proyek oleh tiga organisasi cabang terbesar (PMI, IPMA dan PRINCE 2), dapat dianggap diterima secara umum. Efektivitas manajemen proyek sangat penting dalam memastikan keberhasilan setiap aktivitas.

Berdasarkan beberapa pendapat tersebut dapat

disintesisikan bahwa manajemen proyek software adalah aktivitas mengembangkan software dengan menerapkan pengetahuan dan keterampilan melalui proses perencanaan, penyediaan staf, mengorganisasikan, mengarahkan dan mengontrol sesuai dengan biaya dan penjadwalan waktu yang telah disepakati.

Spectrum Manajemen

Seorang manajer harus paham bahwa perangkat lunak merupakan pekerjaan rekayasa sehingga manusia yang terlibat didalamnya harus memiliki pengetahuan dan keterampilan sesuai bidangnya serta paham tentang manajemen proyek. Seorang manajer harus memiliki kemampuan komunikasi yang komprehensif dengan pelanggan sehingga dapat membangun solusi yang elegan jika terdapat permasalahan. Manajer harus memperhatikan proses yang sedang berjalan dengan metode dan alat teknis yang kompeten dalam mengembangkan perangkat lunak. Manajer harus membuat perencanaan proyek yang solid sehingga produk yang dikembangkan layak dan efektif digunakan. Manajemen proyek software yang efektif berlandaskan 5 P (Orang, Product, Process, dan Project) (Pressman, 2010).

People (Orang)

Dalam sebuah penelitian yang diterbitkan oleh IEEE dengan judul *A field study of the software design process for large systems*, menyatakan bahwa yang berkontribusi paling penting untuk sukses dalam proyek perangkat lunak bukan alat yang digunakan, melainkan memiliki orang-orang yang cerdas dan baik sebagai staf (Curtis et al., 1988). Para manajer berpendapat bahwa orang merupakan faktor yang utama dalam mengembangkan perangkat lunak

(Dangmei, 2017). Pada bagian ini kita akan membahas para pemain yang berpartisipasi dalam proses perangkat lunak dan cara mengorganisir untuk melakukan rekayasa perangkat lunak yang efektif.

Struktural Pelaksana

Aktivitas software dengan structural pelaksana proyek yang dapat dikelompokkan ke dalam lima peran penting pada unsur manajemen (Pressman, 2010):

1. General Manager yang menentukan masalah produk yang biasanya berdampak besar pada proyek.
2. Technical Manager proyek yang bertanggung jawab untuk merencanakan, memotivasi, mengatur, dan mengawasi praktisi perangkat lunak.
3. Praktisi yang menyediakan keterampilan teknis yang diperlukan untuk mengembangkan perangkat lunak baru.
4. Pelanggan dan stockholder yang memiliki kepentingan terhadap hasilnya.
5. Setelah dirilis untuk produksi, pengguna akhir berinteraksi dengan perangkat lunak.

Orang-orang yang termasuk dalam taksonomi ini bertanggung jawab atas semua proyek perangkat lunak. Agar efektif, tim proyek harus diatur sedemikian rupa sehingga memaksimalkan keterampilan dan kemampuan setiap orang. Semuanya itu merupakan tugas dari pemimpin tim.

Pemimpin Tim

Menurut (Weinberg, 2016) menyarankan bahwa seorang pemimpin harus memiliki kemampuan sebagai berikut:

1. Memotivasi orang-orang teknisnya sehingga

menghasilkan kemampuan yang terbaik.

2. Mengorganisasi tim menjadi solid dan percaya diri sehingga membentuk proses yang sudah ada (atau menciptakan proses baru) dan menghasilkan produk final.
3. Memiliki Ide atau inovasi untuk mendorong tim berkreasi dan kreatif sehingga produktivitas menjadi optimal.
4. Kemampuan memecahkan masalah. Manajer proyek perangkat lunak harus berkonsentrasi pada pemahaman masalah yang harus diselesaikan, mengelola aliran ide, dan tetap menjaga kualitas pada saat yang sama.

Perspektif tentang *typical technical manager project* yang efektif menekankan pada empat karakteristik utama (Muller & Turner, 2007):

1. Pemecahan masalah.
Manajer proyek perangkat lunak yang efektif dapat mendiagnosis masalah teknis dan organisasi yang paling relevan secara sistematis dan memotivasi praktisi lain untuk menciptakan solusi. Dapat mengadopsi pengetahuannya yang diperoleh dari pengalaman sebelumnya ke situasi saat ini dan menyesuaikan dalam mengubah kebijakan jika solusi yang dibuat tidak berhasil.
2. Identitas pemimpin.
Manajer proyek perangkat lunak harus bertanggung jawab atas proyek yang dikembangkan dan mampu mengambil kendali saat diperlukan.
3. Pencapaian.
Manajer proyek perangkat lunak harus mampu mengoptimalkan produktivitas, menghargai inisiatif dan pencapaian serta berani mengambil keputusan bahwa pengambilan risiko terkendali tidak akan

dihukum.

4. Pengaruh dan pembangunan tim.

Untuk menjadi manajer proyek perangkat lunak yang efektif, mereka harus mampu memahami sifat orang, memahami sinyal verbal dan nonverbal, dan bereaksi terhadap kebutuhan orang yang mengirimkan sinyal tersebut. Harus mampu mempertahankan kendali dalam situasi stres.

Tim Perangkat Lunak

Penerapan sumber daya dalam proyek yang akan diperlukan X orang yang bekerja selama Z tahun, beberapa opsi di bawah ini dapat disesuaikan (Pressman, 2010).

1. Sebanyak X , Y orang mengerjakan tugas fungsional yang berbeda, yang merupakan kombinasi kerja. Seorang manajer proyek mungkin memiliki enam proyek lain yang harus dia koordinasikan.
2. Sebanyak X orang diminta untuk menyelesaikan tugas fungsional yang berbeda sebanyak Y , sehingga terbentuk "TIM" yang tidak resmi. Tim dapat dibentuk sesuai kebutuhan. Seorang manajer perangkat lunak harus membantu tim dalam bekerja sama.
3. N tim terdiri dari X orang, masing-masing ditugaskan untuk menyelesaikan satu tugas fungsional atau lebih. Setiap tim memiliki struktur khusus yang diterapkan pada semua tim yang bekerja dalam sebuah proyek. Di sini, koordinasi diawasi oleh seorang manajer proyek perangkat lunak dan tim itu sendiri.

Berdasarkan hasil penelitian (Sawyer, 2004) bahwa opsi ke-3 paling produktif karena pendekatan ini setiap tim memiliki struktur yang spesifik dan dikontrol oleh masing-

masing coordinator maupun manajer proyek. Sedangkan menurut (Mantei, 1981) Gaya manajemen organisasi masing-masing dalam menentukan struktur tim yang "terbaik". Mantei (1981) menyarankan secara umum ada tiga organisasi tim yaitu:

1. Desentralisasi demokratis (DD)

Tidak ada pemimpin di tim rekayasa perangkat lunak ini. Sebaliknya, koordinator tugas ditunjuk untuk waktu yang singkat dan kemudian digantikan oleh individu yang memiliki kemampuan untuk mengelola berbagai tugas. Masing-masing kelompok membuat keputusan tentang masalah dan metode yang akan digunakan. Komunikasi antar anggota tim bersifat horizontal.

2. Desentralisasi terkendali (CD)

Tim pengembang harus memiliki kordinator tugas tertentu dan wakil koordinator yang memiliki tanggung jawab untuk subtugas. Pemecahan masalah tetap menjadi aktivitas kelompok masing-masing, tetapi implementasi solusi dibagi di antara subkelompok oleh pemimpin tim. Komunikasi di antara subkelompok dan individu bersifat horizontal. Komunikasi vertical dilakukan sepanjang pengawasan.

3. Terkendali Terpusat (CC)

Pemimpin tim harus mampu memberikan solusi dan mengorganisasi tim internal. Komunikasi antara pemimpin dan anggota tim secara vertikal.

Faktor-faktor yang harus diperhatikan dalam menyusun organisasi tim pengembangan perangkat lunak:

1. Komplexitas masalah yang harus diselesaikan.
2. Size perangkat lunak yang dihasilkan dalam coding.
3. Jadwal yang ditetapkan dalam pelaksanaan
4. Sejauh mana masalah dapat dimodulasi.

5. Kualitas dan keandalan yang dibutuhkan dari perangkat lunak yang dikembangkan.
6. Ketegasan jadwal pengiriman.
7. Kemampuan berkomunikasi yang diperlukan dalam aktivitas.

Tim yang terdesentralisasi menghasilkan solusi yang lebih baik, dan tim semacam itu memiliki kemungkinan yang lebih besar untuk sukses saat bekerja pada masalah yang sulit. Tim CD terpusat untuk pemecahan masalah, sehingga baik struktur tim CD maupun CC dapat berhasil diterapkan pada masalah sederhana. Untuk masalah yang sulit, struktur DD adalah yang terbaik.

Paradigma organisasi untuk tim rekayasa perangkat lunak (Alla et al., 2020):

1. Paradigma tertutup menyusun tim di sepanjang hierarki otoritas tradisional (mirip dengan tim CC). Tim semacam itu dapat bekerja dengan baik ketika memproduksi perangkat lunak yang sangat mirip dengan sebelumnya, tetapi mereka akan cenderung inovatif ketika bekerja dalam paradigma tertutup.
2. Paradigma acak menyusun tim secara longgar dan tergantung pada individu dan inisiatif anggota tim. Ketika inovasi atau terobosan teknologi diperlukan, tim yang mengikuti paradigma acak akan lebih unggul. Namun tim seperti itu mungkin akan kesulitan ketika diperlukan kinerja yang teratur.
3. Meskipun paradigma terbuka berusaha untuk menggabungkan kontrol paradigma tertutup, paradigma acak juga membawa banyak inovasi. Pekerjaan dilakukan dalam tim paradigma terbuka, dengan komunikasi yang baik dan pengambilan keputusan berbasis konsensus. Paradigma terbuka struktur tim sangat membantu untuk menyelesaikan masalah yang rumit, tetapi mungkin tidak efektif

untuk tim lain.

4. Paradigma sinkron bergantung pada kompartementalisasi natural masalah dan mengatur staff proyek untuk mengerjakan elemen masalah dengan sedikit berinteraksi di antara staff. Perlu diingat bahwa organisasi awal tim perangkat lunak adalah organisasi yang terkendali terpusat (CD), yang pada awalnya dikenal sebagai kepala tim programmer.

Struktur ini pertama kali diusulkan oleh Harlan Mills dan dijelaskan oleh Baker. Tim inti terdiri dari seorang insinyur senior (kepala programmer), yang merencanakan, mengkoordinasikan, dan meninjau semua kegiatan teknis tim; staf teknis (biasanya dua sampai lima orang), yang melakukan kegiatan analisis dan pengembangan; dan seorang insinyur cadangan, yang mendukung insinyur senior dalam kegiatannya dan dapat menggantikan insinyur senior. Kepala programmer dapat dibantu oleh satu atau lebih spesialis (misalnya telekomunikasi atau perancang basis data), staf pendukung (misalnya, penulis teknis, staf administrasi teknis, dan personil administrasi), dan pustakawan perangkat lunak. Pustakawan berperan dalam memelihara dan mengontrol semua elemen konfigurasi perangkat lunak (yaitu, dokumentasi, daftar sumber, data, dan media penyimpanan); membantu mengumpulkan dan memformat data; membuat katalog dan mengindeks perangkat lunak; membantu tim dalam penelitian, evaluasi, dan persiapan dokumen (Baker, 2010).

Pustakawan bertindak sebagai pengendali, koordinator, dan berpotensi sebagai evaluator konfigurasi perangkat lunak. Variasi dari tim desentralisasi yang demokratis telah diusulkan oleh Constantine, yang menganjurkan tim dengan kemandirian kreatif. Pekerjaan

perangkat lunak harus memiliki daya tarik, menyalurkan energi kreatif ke dalam kinerja tinggi tim. Untuk mencapai tim berkinerja tinggi adalah sebagai berikut (Alla et al., 2020):

1. Anggota tim harus saling percaya satu sama lain.
2. Distribusi keterampilan harus sesuai dengan masalahnya.
3. Mavericks mungkin harus dikeluarkan dari tim, jika kekompakan tim ingin dipertahankan.

DeMarco dan Lister berpendapat bahwa anggota tim yang bersemangat secara signifikan lebih produktif dan lebih termotivasi, karena memiliki tujuan dan budaya yang sama (DeMarco & Lister, 1999). Namun tidak semua tim bisa menyatu. Faktanya, banyak tim yang menderita apa yang disebut Jackman sebagai "toksisitas tim". Ada lima faktor yang berpotensi merusak lingkungan tim (Jackman, 1998):

1. Suasana kerja yang hiruk pikuk di mana anggota tim membuang energi dan kehilangan fokus pada tujuan pekerjaan yang harus dilakukan.
2. Frustrasi yang tinggi yang disebabkan oleh faktor pribadi, bisnis, atau teknologi yang menyebabkan gesekan di antara anggota tim.
3. Prosedur tidak terkoordinasi dengan baik yang menjadi penghalang pencapaian tujuan.
4. Kurangnya akuntabilitas dan saling menyalahkan.
5. Kegagalan yang terus menerus dan berulang-ulang yang menyebabkan hilangnya kepercayaan diri dan penurunan moral.

Jackman menyarankan yang dapat mengatasi masalah adalah menghindari lingkungan kerja yang hiruk pikuk, manajer proyek harus memastikan bahwa tim memiliki akses ke semua informasi yang diperlukan untuk melakukan pekerjaan sesuai dengan tujuan utama dan

tidak boleh dimodifikasi kecuali benar-benar diperlukan. Selain itu, jika ada informasi yang kurang baik tidak boleh dirahasiakan, melainkan disampaikan kepada tim sedini mungkin.

Proses perangkat lunak yang dipilih secara tidak dapat dihindari dengan dua cara: 1) memastikan bahwa karakteristik perangkat lunak yang akan dibangun sesuai dengan ketelitian dan, 2) mengizinkan tim untuk memilih proses yang telah ditentukan.

Manajer proyek perangkat lunak, yang bekerja sama dengan tim, harus dengan jelas menyempurnakan peran dan tanggung jawab sebelum proyek dimulai. Tim itu sendiri harus menetapkan mekanisme akuntabilitasnya. Setiap tim perangkat lunak pasti mengalami kegagalan-kegagalan kecil. Kunci untuk menghindari suasana kegagalan adalah membangun teknik berbasis tim untuk umpan balik dan pemecahan masalah. Selain itu, kegagalan oleh anggota tim mana pun harus dipandang sebagai kegagalan oleh tim itu sendiri. Hal ini mengarah pada pendekatan yang berorientasi pada tim untuk melakukan tindakan korektif, bukan saling menyalahkan.

Interaksi dan Koordinasi

Ada sejumlah alasan mengapa proyek perangkat lunak mengalami masalah. Pengembangan proyek yang sangat besar sehingga menyebabkan kesulitan yang signifikan dalam mengkoordinasikan anggota tim. Salah satu hal yang sering terjadi adalah ketidakpastian, yang menyebabkan perubahan yang tidak berhenti dan mengganggu tim proyek. Banyaknya sistem harus dapat berinteraksi satu sama lain. Software baru harus berinteraksi dengan software yang sudah ada sehingga system menjadi sinkron. Tipecal perangkat lunak modern

memiliki ketidakpastian, dan interoperabilitas merupakan fakta kehidupan. Tim rekayasa perangkat lunak harus menetapkan metode yang efektif untuk mengoordinasikan orang-orang yang melakukan pekerjaan agar dapat menyelesaikannya secara efektif. Untuk mencapai mekanisme, setiap anggota tim harus berinteraksi secara formal dan informal. Interaksi formal dilakukan melalui "tulisan, pertemuan terstruktur, dan saluran interaksi yang relatif tidak interaktif dan impersonal."(Amrit, 2005). Interaksi informal lebih bersifat pribadi.

Teknik koordinasi proyek dapat dikategorikan sebagai berikut (Amrit, 2005):

1. Pendekatan formal dan impersonal termasuk dokumen rekayasa perangkat lunak dan hasil kerja (termasuk kode sumber), memo teknis, pencapaian proyek, jadwal, dan alat kontrol proyek.
2. Prosedur formal dan interpersonal berkonsentrasi pada proses jaminan kualitas untuk pengembangan perangkat lunak.
3. Prosedur informal dan interpersonal termasuk rapat tim untuk pertukaran data, membuat solusi, dan penyebaran data.

Produk

Seorang manajer proyek perangkat lunak dihadapkan pada dilema di awal proyek rekayasa perangkat lunak. Perkiraan kuantitatif dan rencana yang terorganisir adalah diperlukan, tetapi informasi yang solid tidak tersedia. Analisis terperinci tentang persyaratan perangkat lunak akan memberikan informasi yang diperlukan untuk perkiraan, tetapi analisis sering kali memakan waktu yang lama untuk menyelesaikannya. Lebih buruk lagi, persyaratan berubah seiring proyek berjalan. Oleh karena

itu, sebelum proyek dimulai, kita harus memeriksa produk dan masalah yang ingin ditangani. Ruang lingkup produk harus setidaknya ditetapkan dan dibatasi.

Ruang Lingkup Perangkat Lunak

Proses pertama dalam manajemen proyek perangkat lunak adalah menentukan ruang lingkup perangkat lunak. Ini dilakukan untuk menjawab pertanyaan berikut:

1. Material.
Bagaimana proyek yang akan dikembangkan sesuai dengan proyek yang lebih besar dan batasannya.
2. Perencanaan sistem.
Objektif data apa yang dihasilkan perangkat lunak dan dapat dilihat pelanggan sebagai output.
3. Peran dan kinerja.
Peran apa yang dilakukan aplikasi untuk mengubah data masukan menjadi keluaran dan karakteristik khusus dari perangkat lunak.

Batasan proyek software harus jelas dan dapat dimengerti oleh manajemen dan tingkat teknis.

Proses

Pengembangan dan dukungan adalah fase utama proses perangkat lunak, yang berlaku untuk semua software. Memilih model proses yang tepat untuk perangkat lunak yang akan direkayasa oleh tim proyek adalah masalahnya; berbagai paradigma rekayasa perangkat lunak telah dibahas, termasuk model sekuensial linier, model pembuatan prototipe, model RAD, model incremental, model spiral, model pengembangan berbasis komponen, model pengembangan berspesialisasi, dan model pengembangan berspesialisasi.

Manajer proyek harus memutuskan model proses

mana yang paling tepat untuk (1) pelanggan yang telah meminta produk, (2) karakteristik produk itu sendiri, dan (3) lingkungan proyek tempat tim perangkat lunak bekerja. Ketika model proses telah dipilih, tim kemudian mendefinisikan rencana proyek awal berdasarkan serangkaian kerangka kerja proses umum aktivitas.

Memadukan Produk dan Proses

Desain proyek dimulai dengan perumusan, mengembangkan produk yang diintegrasikan proses. Tim perangkat lunak harus melewati serangkaian aktivitas yang ditetapkan oleh organisasi perangkat lunak sebelum dapat melakukan fungsi apa pun. Perlu dianggap bahwa perusahaan telah melakukan berbagai tindakan sebagai berikut:

1. Interaksi dengan pelanggan. Kegiatan interaksi dengan pelanggan:
 - a. Menyusun daftar masalah.
 - b. Bertemu dengan pelanggan untuk membahas masalah.
 - c. Bersama-sama mengembangkan pernyataan ruang lingkup.
 - d. Tinjau pernyataan ruang lingkup dengan semua pihak terkait.
 - e. Memodifikasi pernyataan ruang lingkup sesuai kebutuhan.
2. Perencanaan kerja.
3. Analisis risiko kerja.
4. Rekayasa aplikasi
5. Konstruksi dan rilis-tugas-tugas yang diperlukan untuk membangun, menguji, menginstal, dan menyediakan dukungan pengguna (misalnya, dokumentasi dan pelatihan).

6. Evaluasi yang diperlukan untuk mendapatkan umpan balik dari pelanggan berdasarkan evaluasi representasi perangkat lunak

Proyek pengembangan perangkat lunak memerlukan tugas-tugas kerja untuk aktivitas interaksi pelanggan sebagai berikut

1. Meninjau permintaan pelanggan.
2. Rencanakan dan jadwalkan rapat rutin dengan pelanggan.
3. Melakukan kajian dalam memberikan solusi dengan pendekatan yang ada.
4. Siapkan file kerja dan jadwal rapat rutin.
5. Melaksanakan rapat.
6. Bersama-sama mengembangkan spesifikasi mini yang mencerminkan data, fungsi, dan fitur perilaku dari perangkat lunak.
7. Tinjau setiap spesifikasi mini untuk kebenaran, konsistensi, dan kurangnya ambiguitas.
8. Kumpulkan mini-spesifikasi ke dalam dokumen pelingkupan.
9. Tinjau dokumen pelingkupan dengan semua pihak terkait.
10. Modifikasi dokumen pelingkupan sesuai kebutuhan.

Proyek

Untuk mengelola proyek perangkat lunak yang sukses, kita harus memahami kesalahan (sehingga masalah dapat dihindari) dan bagaimana melakukannya dengan benar. Dalam sebuah makalah yang sangat bagus tentang proyek perangkat lunak. Sepuluh tanda yang mengindikasikan bahwa proyek sistem informasi dalam bahaya (Sudhakar, 2012):

1. Orang-orang perangkat lunak tidak memahami

kebutuhan pelanggan.

2. Ruang lingkup masalah tidak dijelaskan dengan baik.
3. Manajemen dengan buruk.
4. Tools yang dipilih berubah.
5. Kebutuhan bisnis berubah
6. Tenggat waktu tidak realistis.
7. Pengguna resisten.
8. Sponsor hilang
9. Tidak ada anggota tim proyek yang memiliki kompetensi yang tepat.
10. Manajer dan praktisi tidak mengikuti praktik terbaik.

Pendekatan yang masuk akal dalam mengembangkan proyek perangkat lunak (Sudhakar, 2012):

1. Mulailah dengan langkah yang benar, memahami masalah yang harus diselesaikan, dan kemudian menetapkan tujuan yang realistis untuk setiap orang yang terlibat dalam proyek. Membangun tim yang tepat dan memberikan teknologi, otonomi, dan otoritas.
2. Pertahankan momentum. Untuk menjaga momentum, manajer proyek harus menyediakan insentif untuk menjaga pergantian personel seminimal mungkin, tim harus menekankan kualitas, dan manajemen senior harus melakukan segala cara untuk tidak menghalangi tim.
3. Melacak kemajuan. Untuk proyek perangkat lunak, menilai kemajuan terhadap rata-rata yang dikembangkan.
4. Membuat keputusan yang cerdas.
5. Lakukan analisis postmortem. Tentukan cara yang konsisten untuk belajar dari setiap proyek.

Mengevaluasi rencana dan jadwal aktual, mengumpulkan dan menganalisis metrik proyek perangkat lunak, menerima umpan balik dari anggota tim dan pelanggan, dan mencatat hasilnya.

BAB 4 TEKNIK ESTIMASI BIAYA

Pendahuluan

Estimasi biaya adalah proses menghitung perkiraan biaya yang diperlukan untuk menyelesaikan suatu proyek. Estimasi biaya yang akurat dan realistis sangat penting untuk mengelola anggaran proyek dengan efektif.

Teknik estimasi biaya membantu manajer proyek untuk mengidentifikasi dan menentukan sumber daya yang diperlukan, mengalokasikan anggaran dengan bijaksana, dan memperkirakan biaya yang akan dikeluarkan sepanjang proyek. Estimasi biaya yang baik membantu menghindari risiko kekurangan anggaran, memastikan ketersediaan sumber daya yang tepat pada waktu yang tepat, dan membantu dalam pengambilan keputusan terkait proyek.

Pada umumnya, teknik estimasi biaya melibatkan penggunaan data historis, pemodelan matematis, pengalaman ahli, dan penilaian terperinci dari kebutuhan dan tugas proyek. Metode estimasi biaya dapat bervariasi tergantung pada kompleksitas proyek dan ketersediaan data yang tersedia.

Teknik estimasi biaya juga harus dikombinasikan dengan manajemen risiko yang baik. Identifikasi dan mitigasi risiko yang mungkin mempengaruhi biaya proyek harus diperhitungkan dalam estimasi biaya untuk mengantisipasi kemungkinan perubahan biaya yang tidak terduga.

Estimasi Berbasis Analogi

Estimasi Berbasis Analogi adalah salah satu teknik estimasi biaya yang umum digunakan dalam manajemen proyek. Metode ini melibatkan penggunaan data dari proyek-proyek sebelumnya yang serupa sebagai referensi untuk memperkirakan biaya proyek saat ini. Prinsip dasar di balik estimasi berbasis analogi adalah bahwa proyek-proyek yang serupa dalam hal ukuran, kompleksitas, dan karakteristik lainnya cenderung memiliki pola biaya yang serupa.

Proses estimasi berbasis analogi dimulai dengan mengumpulkan data dari proyek-proyek sebelumnya yang memiliki kesamaan dengan proyek saat ini. Data ini dapat berupa biaya keseluruhan proyek, waktu yang diperlukan, sumber daya yang digunakan, atau parameter lain yang relevan. Selanjutnya, data ini digunakan untuk membandingkan proyek-proyek tersebut dan mengidentifikasi pola atau tren dalam biaya.

Setelah identifikasi pola, pola tersebut diterapkan pada proyek saat ini untuk memperkirakan biaya yang mungkin terjadi. Estimasi ini dilakukan dengan mempertimbangkan faktor-faktor kunci yang mempengaruhi biaya proyek, seperti perbedaan dalam skala atau kompleksitas proyek saat ini dibandingkan dengan proyek sebelumnya. Pemahaman yang baik tentang perbedaan dan kesamaan antara proyek-proyek tersebut penting untuk menghasilkan estimasi yang akurat.

Estimasi berbasis analogi sangat berguna ketika data historis yang relevan tersedia dan proyek saat ini memiliki kesamaan yang signifikan dengan proyek-proyek sebelumnya. Metode ini dapat digunakan dalam berbagai industri dan jenis proyek, termasuk proyek pengembangan perangkat lunak, konstruksi, atau proyek manufaktur.

Namun, penting untuk diingat bahwa setiap proyek memiliki karakteristik unik, dan estimasi berbasis analogi tidak selalu menghasilkan estimasi yang sempurna. Variabilitas dan faktor-faktor khusus dari proyek saat ini harus tetap diperhitungkan, dan estimasi berbasis analogi harus digunakan sebagai panduan awal yang perlu dikonfirmasi dan disesuaikan dengan kondisi proyek yang sebenarnya.

Dengan menggunakan estimasi berbasis analogi dengan hati-hati, manajer proyek dapat memanfaatkan pengetahuan dan pengalaman dari proyek-proyek sebelumnya untuk membuat estimasi biaya yang lebih realistis dan memperbaiki pengambilan keputusan terkait anggaran proyek.

Estimasi Berbasis Bottom-Up

Estimasi Berbasis Bottom-Up adalah salah satu teknik estimasi biaya yang digunakan dalam manajemen proyek. Metode ini melibatkan estimasi biaya untuk setiap komponen individual proyek dan kemudian menggabungkannya menjadi perkiraan biaya keseluruhan. Pendekatan bottom-up memerlukan analisis yang rinci dan penghitungan biaya yang lebih spesifik untuk setiap tugas atau aktivitas dalam proyek.

Proses estimasi berbasis bottom-up dimulai dengan memecah proyek menjadi komponen yang lebih kecil, seperti tugas, aktivitas, atau paket pekerjaan. Setelah itu, estimasi biaya yang terperinci dibuat untuk setiap komponen tersebut. Estimasi ini dapat melibatkan analisis historis, konsultasi dengan anggota tim yang terlibat, atau referensi ke sumber daya eksternal seperti vendor atau kontraktor.

Setelah estimasi biaya individu diperoleh,

komponen-komponen tersebut digabungkan untuk menghasilkan perkiraan biaya keseluruhan proyek. Metode ini memungkinkan manajer proyek untuk memiliki visibilitas yang lebih baik terhadap biaya proyek secara keseluruhan, serta memungkinkan pemantauan dan pengendalian yang lebih baik terhadap setiap komponen biaya.

Kelebihan dari estimasi berbasis bottom-up adalah keakuratan yang lebih tinggi dalam memperkirakan biaya. Dengan memecah proyek menjadi komponen yang lebih kecil, estimasi dapat dilakukan dengan lebih terperinci dan memperhitungkan faktor-faktor unik atau khusus yang mungkin mempengaruhi biaya. Hal ini membantu menghindari ketidakpastian yang mungkin muncul dalam estimasi biaya secara keseluruhan.

Namun, estimasi berbasis bottom-up juga memerlukan upaya dan waktu yang lebih besar untuk dilakukan. Menganalisis setiap komponen proyek dan melakukan estimasi terperinci membutuhkan sumber daya dan tenaga yang cukup. Selain itu, estimasi berbasis bottom-up juga dapat lebih rentan terhadap perubahan atau ketidaktepatan jika komponen-komponen yang diperkirakan tidak dipahami dengan baik atau tidak dianalisis secara menyeluruh.

Dalam kesimpulannya, estimasi berbasis bottom-up memberikan keakuratan yang lebih tinggi dalam memperkirakan biaya proyek dengan memecahnya menjadi komponen yang lebih kecil dan menganalisisnya secara terperinci. Pendekatan ini memungkinkan manajer proyek untuk memiliki visibilitas yang lebih baik terhadap biaya dan membantu dalam pengambilan keputusan terkait anggaran proyek.

Estimasi Berbasis Parametrik

Estimasi Berbasis Parametrik adalah teknik estimasi biaya yang menggunakan model matematis atau statistik untuk menghubungkan variabel-variabel proyek tertentu dengan biaya. Metode ini didasarkan pada asumsi bahwa ada hubungan kuantitatif antara faktor-faktor proyek tertentu dengan biaya yang terkait.

Dalam estimasi berbasis parametrik, parameter atau variabel-variabel yang berpengaruh terhadap biaya proyek diidentifikasi. Contohnya, dalam konteks pengembangan perangkat lunak, parameter seperti jumlah baris kode, kompleksitas fungsional, ukuran tim pengembang, atau durasi proyek dapat digunakan sebagai faktor yang berpengaruh pada biaya.

Selanjutnya, model matematis atau statistik dibuat untuk menggambarkan hubungan antara parameter-parameter tersebut dengan biaya. Model ini dapat berupa persamaan matematis, fungsi regresi, atau rumus yang menggambarkan pola biaya berdasarkan variabel-variabel yang diperhitungkan.

Estimasi berbasis parametrik melibatkan pengumpulan data historis yang relevan untuk mengkalibrasi model dan menguji kecocokannya. Data ini bisa berasal dari proyek-proyek sebelumnya atau sumber data industri yang tersedia. Dengan menggunakan data ini, model dapat disesuaikan dan diterapkan pada proyek saat ini untuk menghasilkan perkiraan biaya yang lebih akurat.

Kelebihan dari estimasi berbasis parametrik adalah kemampuannya untuk memberikan perkiraan biaya yang cepat dan efisien. Dengan menggunakan model matematis atau statistik yang sudah teruji, estimasi dapat dilakukan dengan cepat tanpa perlu melakukan analisis terperinci terhadap setiap komponen proyek.

Namun, estimasi berbasis parametrik memiliki beberapa kelemahan. Pertama, model parametrik membutuhkan data historis yang relevan dan representatif untuk menghasilkan perkiraan yang akurat. Jika data yang tersedia terbatas atau tidak mewakili proyek saat ini dengan baik, estimasi dapat menjadi tidak akurat. Selain itu, estimasi berbasis parametrik tidak selalu mempertimbangkan faktor-faktor unik atau khusus dari proyek tertentu yang dapat mempengaruhi biaya.

Dalam kesimpulannya, estimasi berbasis parametrik adalah teknik estimasi biaya yang menggunakan model matematis atau statistik untuk menghubungkan parameter-parameter proyek dengan biaya. Metode ini dapat memberikan perkiraan biaya yang cepat dan efisien, tetapi membutuhkan data historis yang relevan dan mempertimbangkan keterbatasan dalam memperhitungkan faktor-faktor unik proyek

Estimasi Berbasis Three-Point

Estimasi Berbasis Three-Point, juga dikenal sebagai Estimasi Berbasis Tiga Titik atau Estimasi Berbasis PERT (Program Evaluation and Review Technique), adalah teknik estimasi biaya yang melibatkan penggunaan tiga nilai estimasi untuk setiap aktivitas atau tugas dalam proyek. Metode ini dirancang untuk mengatasi ketidakpastian dan variasi yang mungkin terjadi dalam estimasi biaya.

Dalam Estimasi Berbasis Three-Point, tiga nilai estimasi digunakan: nilai yang paling optimis (O), nilai yang paling pesimis (P), dan nilai yang paling realistis (R). Nilai optimis merupakan perkiraan biaya yang sangat ideal, dengan asumsi bahwa segala sesuatunya berjalan dengan lancar tanpa adanya hambatan atau kendala. Nilai pesimis adalah perkiraan biaya yang paling buruk, dengan

mempertimbangkan kemungkinan terburuk yang mungkin terjadi dalam proyek. Nilai realistis adalah perkiraan biaya yang berdasarkan pada analisis objektif dan pertimbangan rasional atas faktor-faktor yang mungkin mempengaruhi biaya.

Setelah tiga nilai estimasi diperoleh, estimasi biaya yang sebenarnya (E) dihitung menggunakan formula PERT, yang dinyatakan sebagai berikut:

$$E = (O + 4R + P) / 6$$

Rumus ini memberikan bobot yang lebih besar pada nilai realistis dalam perhitungan estimasi, dan memberikan sedikit pengaruh pada nilai optimis dan pesimis. Hal ini dilakukan untuk mencerminkan sifat ketidakpastian dalam proyek dan memberikan estimasi yang lebih seimbang.

Estimasi Berbasis Three-Point membantu dalam mengatasi kesalahan atau bias dalam estimasi biaya yang mungkin terjadi. Dengan mempertimbangkan variasi dalam perkiraan biaya dan memasukkan ketidakpastian ke dalam perhitungan, metode ini memberikan perkiraan yang lebih akurat dan realistis.

Selain itu, Estimasi Berbasis Three-Point juga memungkinkan manajer proyek untuk mengidentifikasi aktivitas-aktivitas yang memiliki tingkat risiko yang lebih tinggi, karena perbedaan yang signifikan antara nilai optimis dan pesimis. Hal ini membantu dalam perencanaan dan pengelolaan risiko proyek dengan lebih baik.

Namun, penting untuk diingat bahwa Estimasi Berbasis Three-Point masih memerlukan penilaian subjektif dan asumsi yang berdasarkan pada pengalaman dan pengetahuan tim proyek. Estimasi ini juga tidak dapat memperhitungkan perubahan-perubahan yang mungkin terjadi selama proyek berlangsung.

Dalam kesimpulannya, Estimasi Berbasis Three-Point adalah teknik estimasi biaya yang menggunakan tiga nilai estimasi (optimis, realistis, dan pesimis) untuk mengatasi ketidakpastian dan variasi dalam estimasi. Metode ini membantu dalam menghasilkan perkiraan biaya yang lebih akurat dan realistis, serta memungkinkan identifikasi risiko yang lebih baik dalam proyek.

Estimasi Berbasis Top-Down

Estimasi Berbasis Top-Down adalah teknik estimasi biaya yang melibatkan pendekatan umum untuk mengestimasi biaya proyek secara keseluruhan berdasarkan pengalaman, pengetahuan, atau data historis yang ada. Metode ini melibatkan pemecahan proyek menjadi bagian-bagian yang lebih besar, seperti fase, modul, atau komponen utama, dan kemudian memberikan perkiraan biaya secara kasar untuk setiap bagian tersebut.

Dalam Estimasi Berbasis Top-Down, estimasi biaya umumnya didasarkan pada persentase dari biaya keseluruhan proyek. Sebagai contoh, manajer proyek mungkin menggunakan persentase tertentu dari biaya total proyek sebelumnya yang memiliki kesamaan atau kompleksitas yang mirip untuk mengestimasi biaya proyek saat ini. Pendekatan ini mengandalkan pemahaman dan pengetahuan yang ada untuk memberikan perkiraan kasar tentang biaya yang mungkin terjadi.

Kelebihan dari Estimasi Berbasis Top-Down adalah kemudahannya dan efisiensinya. Dengan melihat proyek secara keseluruhan dan memberikan perkiraan biaya secara kasar berdasarkan pengalaman atau pengetahuan, estimasi dapat dilakukan dengan cepat dan tanpa perlu analisis terperinci dari setiap komponen proyek.

Namun, Estimasi Berbasis Top-Down juga memiliki

keterbatasan dan risiko ketidakakuratan. Karena estimasi didasarkan pada persentase atau perkiraan umum, detail dan variasi dalam setiap bagian proyek mungkin tidak diperhitungkan secara tepat. Jika ada perbedaan signifikan dalam kompleksitas atau persyaratan proyek saat ini dibandingkan dengan proyek sebelumnya yang digunakan sebagai acuan, estimasi berbasis top-down mungkin tidak memberikan hasil yang akurat.

Penting untuk dicatat bahwa Estimasi Berbasis Top-Down sering digunakan pada tahap awal perencanaan proyek untuk memberikan pandangan umum tentang biaya. Estimasi ini perlu diperbarui dan disesuaikan secara bertahap saat proyek berkembang dan informasi yang lebih rinci tersedia.

Dalam kesimpulannya, Estimasi Berbasis Top-Down adalah teknik estimasi biaya yang memberikan perkiraan kasar berdasarkan pengalaman, pengetahuan, atau data historis yang ada. Pendekatan ini memungkinkan estimasi yang cepat dan efisien, tetapi harus digunakan dengan hati-hati dan disesuaikan seiring dengan perkembangan proyek yang lebih rinci.

Estimasi Berbasis Expert Judgment

Estimasi Berbasis Expert Judgment adalah teknik estimasi biaya yang mengandalkan penilaian dan pendapat ahli dalam bidang terkait untuk mengestimasi biaya proyek. Metode ini mengumpulkan dan memanfaatkan pengetahuan dan pengalaman ahli untuk memberikan perkiraan biaya yang lebih akurat.

Dalam Estimasi Berbasis Expert Judgment, ahli yang terlibat dalam proyek atau memiliki pengetahuan dan pengalaman yang relevan dalam bidang yang bersangkutan, seperti manajer proyek, spesialis domain, atau konsultan,

diminta untuk memberikan perkiraan biaya berdasarkan penilaian mereka. Ahli ini dapat mengambil pendekatan yang berbeda-beda, seperti menggunakan pengalaman mereka sebelumnya, menerapkan metode perbandingan atau analogi, atau mempertimbangkan faktor-faktor khusus yang mungkin mempengaruhi biaya.

Kelebihan dari Estimasi Berbasis Expert Judgment adalah menggabungkan pengetahuan dan pengalaman dari mereka yang memiliki pemahaman mendalam tentang proyek atau bidang yang bersangkutan. Dengan memanfaatkan penilaian ahli, estimasi biaya dapat menjadi lebih terperinci dan mengambil faktor-faktor penting yang mungkin terlewatkan dalam metode estimasi lainnya.

Namun, Estimasi Berbasis Expert Judgment juga memiliki keterbatasan. Perkiraan biaya yang didasarkan pada penilaian subjektif ahli masih rentan terhadap bias atau penilaian yang tidak akurat. Selain itu, mengumpulkan pendapat ahli mungkin membutuhkan waktu dan usaha yang signifikan, terutama jika proyek melibatkan banyak aspek yang kompleks.

Penting untuk melibatkan ahli yang berkualifikasi dan berpengalaman dalam estimasi berbasis expert judgment dan memastikan bahwa mereka memiliki pemahaman yang komprehensif tentang proyek dan faktor-faktor yang mempengaruhi biaya. Selain itu, estimasi berbasis expert judgment perlu didukung dengan data historis atau informasi lain yang relevan untuk meningkatkan keakuratan estimasi.

Estimasi Berbasis Pemilihan Vendor

Estimasi Berbasis Pemilihan Vendor adalah teknik estimasi biaya yang melibatkan pemilihan vendor atau pihak ketiga yang akan terlibat dalam proyek dan meminta

mereka untuk mengajukan proposal biaya. Metode ini digunakan ketika organisasi mempertimbangkan untuk mengontrak atau menggandeng vendor eksternal untuk menyediakan layanan atau produk tertentu dalam proyek.

Dalam Estimasi Berbasis Pemilihan Vendor, langkah-langkah umumnya meliputi:

1. Identifikasi kebutuhan

Organisasi harus mengidentifikasi kebutuhan spesifik proyek yang dapat dilakukan oleh vendor. Ini melibatkan memahami persyaratan, fungsi, dan lingkup pekerjaan yang harus dilakukan oleh vendor.

2. Seleksi vendor

Organisasi melakukan proses pemilihan vendor dengan mengirimkan permintaan proposal (Request for Proposal/RFP) atau melalui proses penawaran yang terstruktur. Vendor-vendor yang berpotensi diundang untuk mengajukan proposal yang mencakup rincian biaya untuk pekerjaan yang diminta.

3. Evaluasi proposal

Organisasi mengevaluasi proposal yang diterima dari vendor-vendor yang berpartisipasi. Evaluasi dapat meliputi penilaian kualitas, keahlian, pengalaman, dan reputasi vendor, serta analisis biaya yang diajukan.

4. Negosiasi kontrak

Setelah evaluasi proposal, organisasi dapat melakukan negosiasi dengan vendor terpilih untuk mencapai kesepakatan yang memadai dalam hal biaya dan lingkup pekerjaan yang akan dilakukan. Pada tahap ini, estimasi biaya dapat direvisi dan disesuaikan berdasarkan hasil negosiasi.

Kelebihan dari Estimasi Berbasis Pemilihan Vendor adalah bahwa estimasi biaya didasarkan pada proposal

yang diajukan oleh para ahli atau vendor yang berpengalaman dalam bidangnya. Vendor-vendor tersebut dapat memberikan perkiraan biaya yang lebih akurat berdasarkan pengetahuan mereka tentang layanan atau produk yang mereka tawarkan.

Namun, penting untuk diingat bahwa estimasi biaya yang diberikan oleh vendor masih dapat berisi faktor-faktor yang tidak terlihat atau tidak diungkapkan secara rinci. Oleh karena itu, organisasi harus melakukan evaluasi dan negosiasi yang cermat untuk memastikan bahwa estimasi biaya mencerminkan kebutuhan dan persyaratan proyek dengan tepat.

BAB 5 MANAJEMEN WAKTU PROYEK PERANGKAT LUNAK

Pendahuluan

Pada beberapa proyek perangkat lunak diperlukan langkah dalam mendefinisikan dan menjadwalkan kegiatan, memperkirakan sumber daya manusia, memperkirakan durasi pengerjaan proyek, dan mengembangkan model pada saat proyek berlangsung. Sebuah penelitian menyatakan bahwa keterlambatan pengiriman adalah salah satu masalah yang paling sering mempengaruhi proyek dengan hampir 47%. Salah satu penyebabnya adalah penerapan manajemen waktu proyek yang buruk. Akibat dari keterlambatan proyek berdampak pada semua pemangku kepentingan yang terlibat dalam proyek. Oleh sebab itu diperlukan teknik dalam mengelolanya waktu dalam proyek atau sering disebut dengan manajemen waktu proyek atau *time management project*.

Manajemen waktu proyek dimulai sejak awal inisiasi proyek dengan mengidentifikasi durasi proyek yang dibutuhkan dan pencapaian milestones, sebelum mendapatkan jadwal terperinci selama fase perencanaan (Hazar H, 2014). Manajemen waktu proyek adalah manajemen waktu yang digunakan pada kemajuan dalam kegiatan proyek berlangsung hingga selesai. Manajemen waktu yang baik membutuhkan perencanaan, penjadwalan, pemantauan, dan pengendalian semua aktivitas proyek. Manajemen waktu adalah salah satu dari enam fungsi

utama manajemen proyek, berdasarkan Project Management Institute (PMI). The Project Management Institute (PMI) membuat Project Management Fundamentals Guide (PMBOK® Guide), sebuah standar yang diakui dalam profesi manajemen proyek yang menyediakan pedoman, teknik dan alat untuk manajemen proyek (PMI, 2008).

Pada proses manajemen waktu proyek pada umumnya mengacu pada cara manajer proyek mempersiapkan strategi untuk mengalokasikan waktu yang tepat untuk setiap tugas dan memastikan tenggat waktu pada setiap tahap proyek hingga proyek itu diserahkan atau delivery project.

Tujuan Manajemen Waktu

Manajemen waktu proyek bertujuan agar proses pelaksanaan proyek sesuai dengan jadwal yang sudah ditetapkan sehingga proyek dapat selesai sesuai dengan target waktu yang disepakati bersama dengan stakeholder.

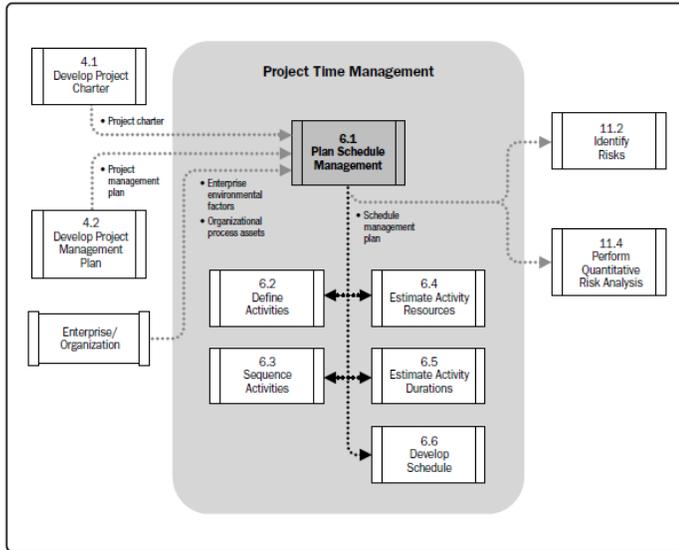
Mekanisme Manajemen Waktu

Pada mekanismendalam manajemen waktu proyek terdapat beberapa fase dalam pelaksanaan manajemen waktu proyek yaitu sebagai berikut:

1. Manajemen rencana jadwal (*schedule management plan*)

Manajemen rencana jadwal adalah proses dalam menetapkan suatu kebijakan, prosedur dan dokumentasi untuk mengelola, membangun, dan mengimplementasikan dan mengontrol waktu pelaksanaan proyek. Manfaat dari proses ini adalah menyediakan petunjuk tentang bagaimana waktu proyek akan dikelola (PMI, 2013). Diagram alir pada

proses ini dapat ditunjukkan pada Gambar 3.



Gambar 3. Diagram alir plan schedule management (sumber: PMBOK Fifth Edition by PMI)

Pada gambar 3 menunjukkan diagram alir dari proses manajemen rencana jadwal. Pada proses ini diperlukan input, alat dan teknik sehingga menghasilkan output yaitu schedule management plan. Selain itu proses ini juga berkaitan dengan proses selanjutnya seperti define activities, sequence activities dan lain sebagainya. Input pada proses ini yaitu

- a. Project management plan

Pada project manajemen plan mencakup scope baseline yang menjelaskan terkait dengan ruang lingkup proyek. Selain itu terdapat *work breakdown structure* (WBS) yang menjelaskan terkait detail dari aktivitas, durasi dan jadwal dalam pengerjaan proyek.

- b. Project charter

Project charter menjelaskan terkait milestone dari jadwal proyek dan kebutuhan proyek yang berpengaruh terhadap pelaksanaan proyek.

- c. Enterprises environmental factors
Faktor lingkungan perusahaan atau enterprises environmental factors mencakup budaya dan struktur organisasi perusahaan, ketersediaan sumber daya dan keterampilan, software pendukung dalam pengelolaan jadwal proyek seperti micosoft project, asana dan lainnya serta kebijakan dan aturan yang diterapkan oleh perusahaan.
- d. Organizational process assets
Pada aset organisasi proses ini mencakup software untuk monitoring, kontrol dan pelaporan, informasi history perusahaan, kebijakan maupun prosedur dalam mengkontrol jadwal pelaksanaan proyek, petunjuk penyelesaian proyek, dan prosedur dalam mengkontrol risiko.

Sedangkan untuk alat dan teknik yang digunakan dalam plan schedule manajemen yaitu

- a. Expert judgment
Expert judgement biasanya dilakukan secara focus group disccussion (FGD) oleh pemangku kepentingan dalam proyek tersebut untuk mendapatkan informasi terkait informasi detail proyek yang diharapkan, kondisi lingkungan perusahaan, dan sumber daya yang tersedia.
- b. Analytical techniqe
Metode yang menganalisis masalah, fakta, atau status dalam pelaksanaan proyek

sehingga dapat memperkirakan hasil secara akurat. Metode ini juga digunakan untuk memecahkan masalah tertentu dalam tugas tertentu.

c. Meetings

Team proyek dapat melakukan pertemuan untuk membuat manajemen rencana jadwal. Pemangku kepentingan yang terlibat yaitu manajer proyek, anggota team proyek, dan personal yang terlibat dalam pelaksanaan proyek

Untuk output yang diharapkan dari proses ini adalah adanya rencana maupun jadwal untuk melakukan monitoring dan kontrol terhadap pelaksanaan proyek secara detail. Beberapa point output yang dihasilkan yaitu

a. *Project schedule mode development*

Pada output ini, diharapkan dapat mendapatkan metodologi dalam pembuatan jadwal pelaksanaan proyek dan software pendukung yang digunakan seperti microsoft project, asana, dan lain sebagainya.

b. *Level of accuracy*

Output ini menjelaskan tingkat atau range yang realistis dalam proses penjadwalan setiap fase dalam proyek. Contoh dalam progress pelaksanaan proyek apabila memiliki tingkat rendah yaitu 0%-40%, sedang 41%-79%, tinggi yaitu 80% -100%.

c. *Unit of measure*

Output ini memberikan informasi satuan dalam tolak ukur proses pelaksanaan proyek. Contoh banyaknya waktu pengerjaan oleh team proyek diukur dalam satuan jam, hari,

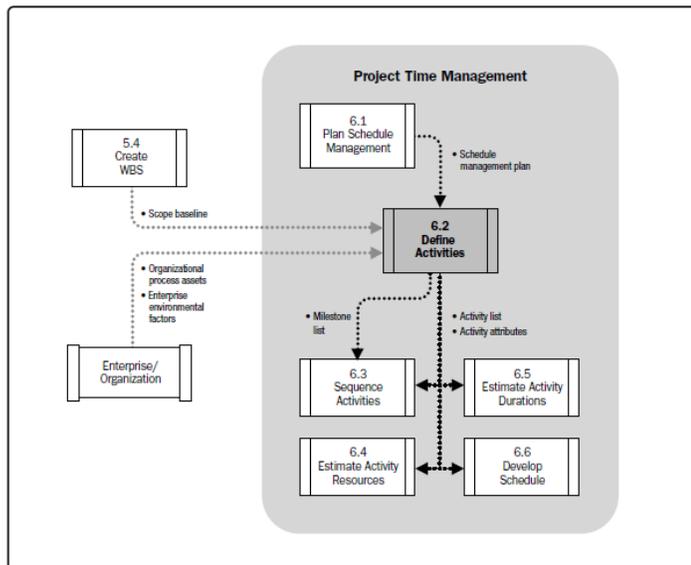
atau minggu.

d. *Process descriptions*

Output ini memberikan informasi terkait deskripsi dari setiap proses manajemen jadwal yang harus terdokumentasi dengan baik.

2. Menentukan aktivitas (*define activities*)

Proses ini bertujuan untuk mengidentifikasi dan mendokumentasikan kegiatan yang spesifik agar proyek dapat terselesaikan. Proses ini juga bermanfaat agar aktifitas dapat di break down atau dijabarkan sesuai dengan dasar dalam pelaksanaan proyek seperti estimation, scheduling, executing, monitoring dan controlling (PMI, 2013). Alur diagram dalam proses define activities ini ditunjukkan pada Gambar 4.



Gambar 4. Diagram Alir Define Activities (sumber: PMBOK Fifth Edition by PMI)

Pada gambar 4 merupakan alur dalam membuat

define activities yang membutuhkan input dari scope baseline, organizational process assets, enterprises environmental factors, dan schedule management plan. Kemudian output yang dihasilkan yaitu milestone list, activity list dan activity attributes. Proses ini juga berkaitan dengan proses lainnya yaitu sequence activities, estimate activity durations dan lainnya. Proses ini hampir sama dengan schedule management plan, namun terdapat perbedaan pada alat dan tekniknya serta output yang dihasilkan. Berikut adalah penjelasan dari input dari proses define activities:

- a. *schedule management plan*
- b. *scope baseline*
work breakdown structure (WBS), goals dari proyek, dan informasi kendala yang terdokumentasi dalam scope baseline diperlukan dalam proses define activities.
- c. *Enterprise environmental factors*
- d. *Organizational process assets*

Sedangkan pada alat dan teknik yang digunakan pada proses ini adalah

- a. *Decomposition*
Teknik ini membagi menjadi sub-sub bagian dari ruang lingkup proyek sehingga lebih mudah untuk di kelola. Teknik ini dapat berupa list aktifitas yang dirangkum dalam WBS sehingga hasil yang didapatkan lebih akurat.
- b. *Rolling Wave Planning*
Teknik ini merupakan perencanaan yang berulang yang mana pekerjaan harus diselesaikan dengan waktu dekat dan secara rinci pada setiap fasenya. Hal ini berlaku

untuk setiap level pekerjaan dalam pelaksanaan proyek. Sehingga proyek lebih terkontrol dengan baik

c. *Expert Judgment*

Beberapa output pada proses define activities yaitu

a. *Activity list*

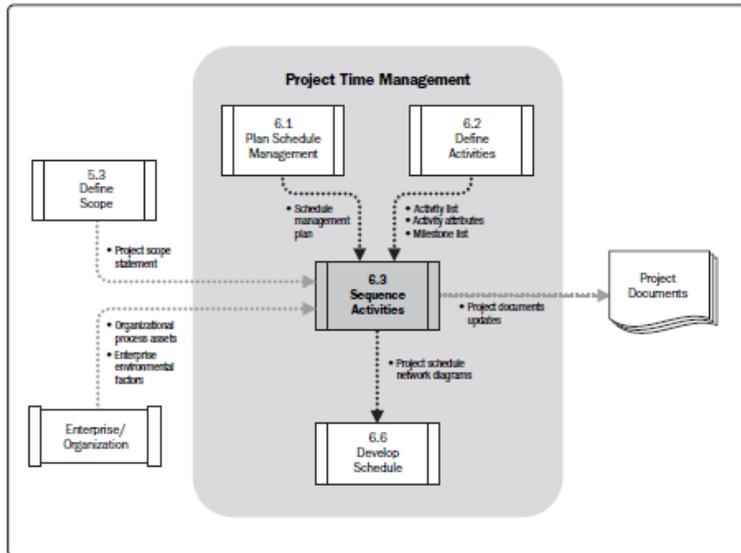
Activity list merupakan daftar lengkap yang mencakup semua aktivitas jadwal yang diperlukan dalam pelaksanaan proyek. Output ini juga memberikan informasi terkait aktifitas-aktifitas secara detail yang dilakukan yang sesuai dengan ruang lingkup proyek. Hal ini bertujuan anggota tim proyek dapat memahami setiap pekerjaan yang harus diselesaikan.

b. *Milestone list*

Milestone merupakan titik atau peristiwa-peristiwa penting dalam sebuah proyek. *Milestone list* adalah daftar dari milestone yang menunjukkan apakah dalam setiap titik atau peristiwa tersebut suatu keharusan ataupun hanya sebagai opsional.

3. Mengurutkan aktivitas (*sequence activities*)

Proses mengidentifikasi dan mendokumentasikan hubungan antara kegiatan proyek. Manfaat utama dari proses ini adalah mendefinisikan urutan logis dari pekerjaan untuk mendapatkan efisiensi yang besar dan terhindar dari kendala proyek.



Gambar 5. Diagram alir sequence activities (sumber: PMBOK Fifth Edition by PMI)

Pada gambar 5 menunjukkan alur dari input dan output dari proses sequence activities. Selain itu juga berkaitan dengan beberapa proses lainnya seperti develop schedule, project documents dan lainnya.

Input dari sequence activities dapat dilakukan dengan beberapa proses yaitu

- a. Schedule management plan
- b. Activity list
- c. Milestone list
- d. Project scope statement
- e. Enterprise environment factors
- f. Organizational process assets

Pada proses input sequence activities merupakan hasil dari proses sebelumnya yang telah dilakukan seperti activity list dan milestone list adalah output dari proses define activities sedangkan schedule management plan, project statement, enterprise

environment factors dan organizational process assets merupakan bagian dari proses sebelumnya yang telah dijelaskan.

Untuk alat dan teknik yang digunakan pada proses ini adalah

a. *Precedence Diagramming Method (PDM)*

Teknik yang digunakan untuk membangun model jadwal di mana aktivitas diwakili oleh node dan secara grafis dihubungkan oleh satu atau lebih hubungan untuk menunjukkan urutan kegiatan yang akan dilakukan. Activity-on-node (AON) adalah salah satu metode untuk merepresentasikan PDM. AON pada umumnya metode yang digunakan oleh sebagian besar manajemen proyek perangkat lunak.

b. *Dependency Determination*

Ketergantungan atau Dependency memiliki karakteristik atribut berikut: wajib (mandatory) atau bebas (discretionary), internal atau eksternal. Dependency memiliki empat atribut, tetapi dua dapat diterapkan sekaligus dalam mengikuti cara: mandatory external dependencies, mandatory internal dependencies, discretionary external dependencies, atau discretionary internal dependencies.

1) *Mandatory dependencies*

Tim proyek menentukan dependensi mana yang wajib selama proses pengurutan aktivitas pelaksanaan proyek. Mandatory Dependency harus dimasukkan dalam penetapan jadwal pelaksanaan proyek.

2) Discretionary dependencies.

Tim proyek dapat menentukan dependensi mana yang bebas selama proses pengurutan aktivitas pelaksanaan proyek.

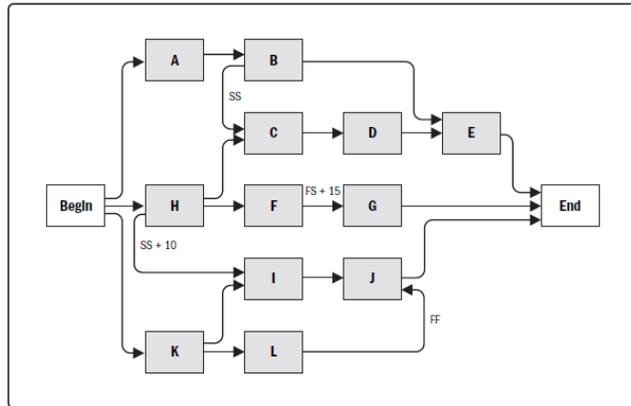
c. Leads and Lags

Prospek atau leads merupakan jumlah waktu yang dapat dimajukan oleh penerus aktivitas sketika proses aktivitas sebelumnya yang dilakukan oleh tim pelaksana sebelumnya belum selesai dilakukan. Seperti contohnya ketika dalam membangun aplikasi, proses implementasi yang dilakukan oleh tim programmer dapat dimajukan 2 minggu lebih cepat dari jadwal ketika tim desain sedang membuat desain mockup tampilan aplikasi. Sedangkan jeda atau lags adalah jeda waktu yang dibutuhkan oleh penerus aktivitas proyek untuk dapat melaksanakan aktivitas sebelum tim aktivitas sebelumnya menyelesaikan pekerjaannya. Seperti contohnya pada proses testing aplikasi dapat dilakukan setelah 1 bulan aktivitas coding selesai.

Hasil atau output dari proses sequence activities adalah sebagai berikut

a. *Project schedule network diagrams*

Diagram jaringan jadwal proyek merupakan representasi grafis dari hubungan logis diantara aktivitas jadwal proyek. Contoh diagram dapat ditunjukkan pada gambar 6.



Gambar 6. Project schedule network diagrams
(sumber: PMBOK Fifth Edition by PMI)

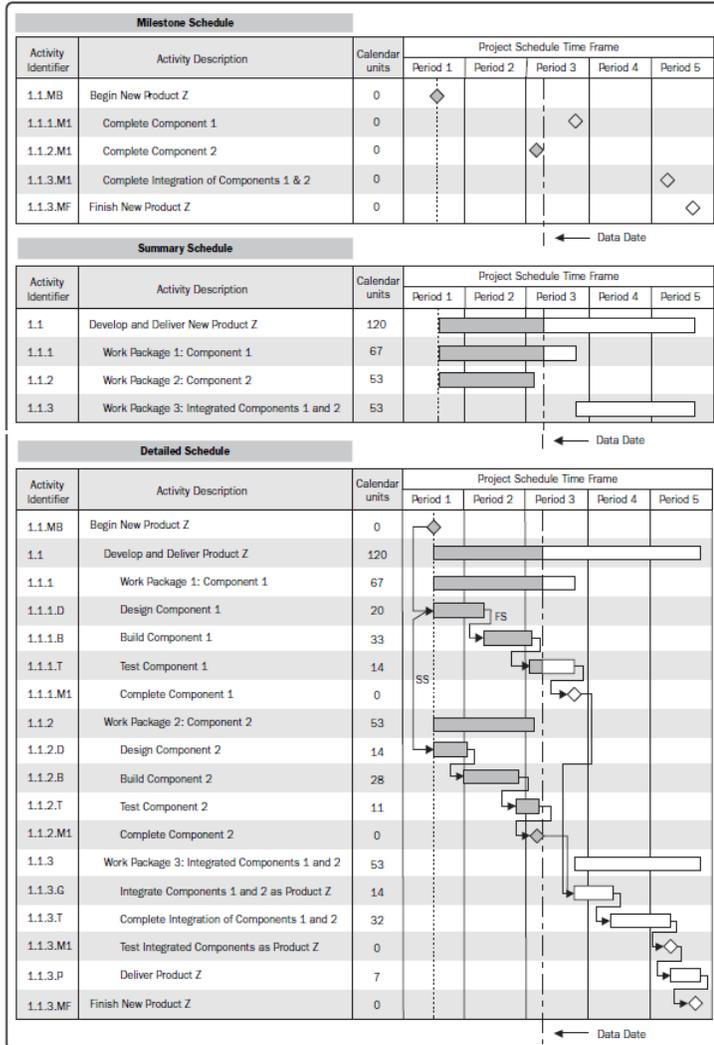
Pada gambar 6 menunjukkan alur diagram dari mulai hingga akhir dari proyek. Pembuatan alur diagram ini dapat menggunakan manual atau aplikasi pendukung seperti Microsoft Visio atau yang lainnya. Dalam penulisan aktivitas dapat ditulis secara diringkas. Selain itu diagram tersebut menggambarkan urutan aktivitas.

4. Membuat Jadwal (*Develop Schedule*)

Membuat jadwal atau develop schedule adalah proses menganalisis urutan aktivitas, durasi, kebutuhan sumber daya, dan batasan jadwal untuk membuat model jadwal proyek. Manfaat utama dari proses ini adalah dengan memasukkan aktivitas, durasi, sumber daya, ketersediaan sumber daya, dan hubungan logis. Hal ini akan menghasilkan model jadwal dengan tanggal yang direncanakan untuk menyelesaikan aktivitas proyek atau disebut dengan *project schedule*.

Project schedule atau jadwal proyek menyajikan aktivitas terkait dengan tanggal, durasi, milestone, dan sumber daya yang direncanakan. Jadwal proyek

dapat mencakup tanggal mulai yang direncanakan dan tanggal selesai yang direncanakan untuk setiap kegiatan. Berikut adalah contoh dari jadwal proyek yang telah dibuat dalam proyek yang ditunjukkan pada gambar 7.



Gambar 7. Contoh Jadwal proyek

Pada gambar 7 menunjukkan pemetaan dari jadwal

untuk contoh proyek yang sedang dilaksanakan. Setiap aktivitas yang sedang berjalan memiliki lama waktu dan status yang dibutuhkan. Selain itu juga terdapat milestones sebagai gambaran berapa lama proyek akan selesai. Kemudian ringkasan dari aktivitas proyek yang akan dilaksanakan dan terakhir adalah rincian aktivitas beserta waktu dan status yang sedang berjalan.

Pada proses manajemen waktu pada proyek yang dibahas pada buku ini masih terdapat beberapa proses lagi yang tidak dapat dijelaskan secara rinci yaitu estimate activity resource, estimate activity durations, develop schedule dan control schedule.

BAB 6 MANAJEMEN MUTU PROYEK PERANGKAT LUNAK

Pendahuluan

Dalam sebuah proyek perangkat lunak, mutu adalah hal yang sering di perbincangkan dan perdebatkan di setiap implementasi proyek perangkat lunak tersebut. Mutu dari proyek perangkat lunak dan hasil implementasi yang dilakukan menjadi acuan serta referensi untuk proyek-proyek perangkat lunak selanjutnya.

Secara konvensional, karakteristik hasil dari sebuah proyek adalah mutu yang berupa keunggulan, kemudahan dalam penggunaan dan perawatan, serta kebagusan hasil sebuah proyek. Dari sisi tim pelaksana proyek, mutu berupa kesesuaian hasil dengan standar yang telah disepakati. Sedangkan dari sisi pelanggan mutu berupa ketercapaiannya kebutuhan atas harapan pelanggan kepada hasil sebuah proyek.

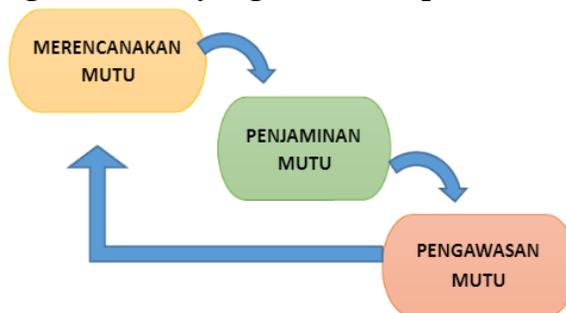
Ketika membicarakan mutu maka pemahaman ISO (*International organization For Standardization*) mendefinisikan kualitas sebagai “*the degree to which a set of inherent characteristics fulfils requirements*” (ISO9000:2000) harus dipahami terlebih dahulu. Dengan dasar ISO ini pula maka penerapan di dalam manajemen mutu proyek perangkat lunak juga mengambil beberapa prinsip-prinsip dari mutu yang harus dijalankan di dalam setiap tahapan proyek perangkat lunak di dalam perusahaan.

Para ahli mengartikan mutu sebagai kecocokan

terhadap kebutuhan (*conformance to requirements*), yaitu prosedur dan hasil proyek perangkat lunak sesuai dengan kriteria dan standar yang disepakati secara tertulis, serta kecocokan untuk pemakaian (*fitness dor use*) yaitu kesesuaian antara pelanggan dan tim proyek, maka dapat menjadi suatu kesepakatan bersama mengenai standar yang dapat memenuhi kebutuhan dan harapan tim pelaksana proyek serta pelanggan.

Manajemen Mutu Proyek Perangkat Lunak

Manajemen mutu adalah suatu pekerjaan yang dikerjakan guna memelihara level mutu yang dikehendaki oleh perusahaan. Pekerjaan ini meliputi runtutan aktivitas pekerjaan seperti memastikan standar mutu, aturan yang dibutuhkan, dan cakupan lainnya yang dapat memastikan mutu hasil sebuah proyek perangkat lunak. Jaminan mutu secara berkala mengevaluasi pelaksanaan proyek secara menyeluruh untuk memastikan bahwa pengerjaan dan hasil proyek memenuhi standar yang disepakati bersama. Serta kendali mutu untuk memantau hasil proyek secara spesifik, guna memastikan bahwa hasil proyek tersebut sesuai dengan standar yang sudah disepakati bersama.



Gambar 8. Tahapan manajemen mutu proyek

Proses manajemen mutu proyek dibagi menjadi 3 tahapan utama seperti pada gambar nomor 8 yaitu:

1. Perencanaan mutu (Quality Planning)
2. Jaminan mutu (Perform Quality Assurance)
3. Kendali mutu (Perform Quality Control)

Tahapan Manajemen Mutu

Perencanaan Mutu (Quality Planning)

Perencanaan mutu yang diinginkan dari hasil proyek perangkat lunak tersebut dijalankan untuk mencukupi kebutuhan bisnis dan target manajemen proyek. Perencanaan mutu ini didetailkan pada beberapa kriteria yang direncanakan harus dicapai dari setiap fase atau hasil produk perangkat lunak yang diimplementasikan perusahaan. kesepakatan dan persetujuan dari manajemen puncak mengenai perencanaan mutu harus ditetapkan di awal dan sebelum proyek berjalan. Pada tahap perencanaan mutu ini penting untuk mencegah kesalahan dengan cara:

1. Menetapkan bahan baku yang sesuai.
2. Mengajarkan dan menegakkan pemahaman tentang mutu.
3. Merancang aktivitas yang mendukung hasil yang sesuai.

Langkah-langkah untuk menyusun rencana mutu sebagai berikut:

1. Definisikan mutu yang diharapkan dari deliverables proyek. Berikan Batasan-batasan deliverables proyek sesuai permintaan yang telah ditetapkan dalam lingkup proyek.
2. Tentukan kriteria pemenuhan mutu dan standar pencapaiannya. Definisi mutu yang telah disebutkan harus memenuhi kriteria sesuai permintaan.
3. Definisikan proses yang dibutuhkan untuk mencapai tujuan mutu. Sesudah kriteria mutu dan standar pencapaiannya sudah ditentukan, selanjutnya

mendefinisikan bagaimana proses pelaksanaan yang sesuai standar untuk memenuhi kriteria tersebut.

4. Deskripsikan quality assurance dan cara melakukan kontrol mutu. Saat proses dilaksanakan, kontrol perlu dilakukan agar tetap sesuai standar mutu, dan dilakukan quality assurance terhadap deliverables apakah memenuhi kriteria mutu.

Di dalam merencanakan mutu kita juga butuh cara dalam mewujudkannya, salah satu cara yang bisa kita gunakan adalah teknik desain eksperimen. Desain eksperimen ini adalah teknik dalam merencanakan mutu yang dapat membantu identifikasi variable yang memiliki dampak besar kepada hasil keseluruhan dari sebuah proses. Teknik desain eksperimen ini juga berlaku pada beberapa hal yang ada pada manajemen proyek, seperti biaya dan jadwal. Pada penggunaan teknik tersebut perlu didokumentasikannya faktor-faktor penting yang secara langsung berkontribusi untuk memenuhi kebutuhan atau permintaan pelanggan. Pimpinan proyek adalah orang yang paling bertanggungjawab terhadap mutu sebuah proyek.

Ada beberapa cakupan aspek di dalam proyek perangkat lunak yang perlu diperhatikan pada proses perencanaan mutu proyek tersebut. Cakupan aspek proyek perangkat lunak yang perlu diperhatikan sebagai berikut:

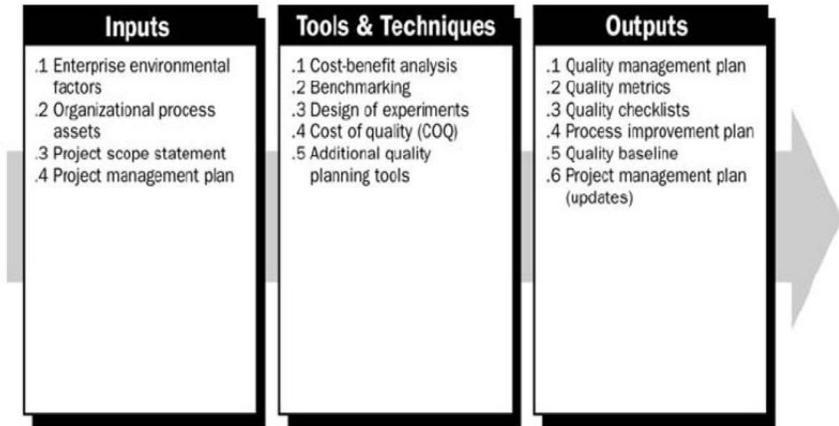
1. Fungsi
Ketepatan hasil proyek dengan fungsi utama hasil proyek itu sendiri atau biasa disebut ciri khas operasi dari suatu produk.
2. Fitur
Ciri khas produk yang membedakan dari produk lain merupakan karakteristik pelengkap yang mampu menimbulkan kesan yang baik bagi pengguna.
3. Sistem output
Monitor atau layar yang menghasilkan sebuah

laporan sebagai hasil akhir dari system yang berjalan.

4. Cakupan kerja
Sampai mana produk hasil proyek tersebut bisa digunakan dengan baik oleh pelanggan.
5. Reliability
Kebutuhan pelanggan terhadap hasil proyek karena kehandalannya atau karena kemungkinan rusaknya rendah.
6. Pemeliharaan
Kemudahan produk hasil proyek jika diperbaiki atau kemudahan memperoleh komponen produk tersebut.

Dari semua proses aktivitas perencanaan mutu yang sudah dilakukan maka akan ada keluaran seperti berikut dan tergambar pada gambar 9:

1. Quality management plan yaitu perencanaan manajemen mutu perangkat lunak.
2. Quality metrics yaitu matrix pengukuran proyek perangkat lunak, dari tingkat kegagalan dan tingkat penyelesaian.
3. Quality checklist yaitu daftar pemeriksaan mutu preangakat lunak.
4. Process improvement plan yaitu perencanaan proses pengembangan.
5. Quality baseline yaitu standar mutu dari proyek.
6. Project management plan (update) yaitu manajemen perencanaan proyek perangkat lunak.



Gambar 9. Kerangka perencanaan manajemen mutu mutu proyek perangkat lunak

Penjaminan Mutu (Quality Assurance)

Penjaminan atau evaluasi mutu meliputi langkah evaluasi pelaksanaan proyek perangkat lunak disetiap fase prosedur pelaksanaannya baik dari sisi software, hardware maupun brainware (tim pelaksana sampai pemakai sistem). Pelaksanaan dari hasil mutu harus selalu dinilai oleh pimpinan proyek dan biasanya di dalam sebuah proyek terdapat petugas pengawas mutu secara terpisah. Hal tersebut dijalankan karena untuk menjaga tingkat objektifitas dari hasil yang diinginkan dari suatu proyek perangkat lunak.

Dari aktivitas penjaminan mutu ini maka akan ada keluaran sebagai berikut:

1. *Requested changes*

Permintaan berubah karena untuk meningkatkan efektifitas dan efisiensi dari peraturan, tahapan dan prosedur dalam proyek, sehingga memberikan keuntungan lebih banyak bagi setiap stakeholder.

2. *Recommended corrective action*

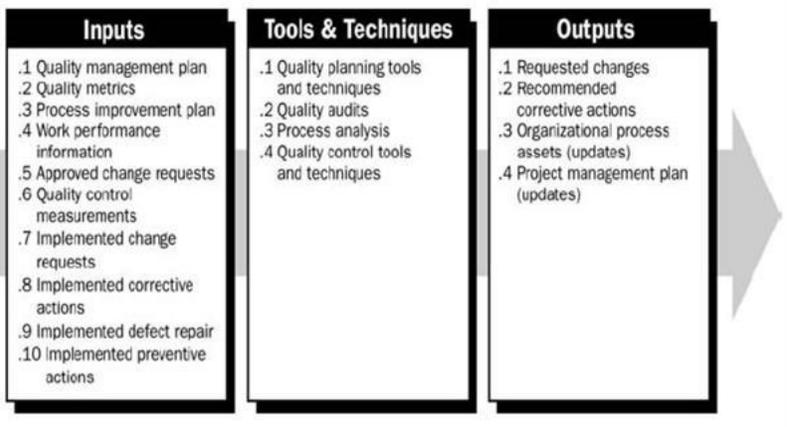
Uraian lebih detail berdasarkan requested changes. Revisi atas ketentuan mutu dari proyek perangkat lunak ini dapat dijadikan acuan untuk tindakan perbaikan untuk kasus-kasus terkait dengan mutu dari proses dan hasil yang diharapkan dari proyek perangkat lunak.

3. *Organizational process assets*

Hasil dari setiap proyek perangkat lunak harus dapat dikontribusikan terhadap penambahan nilai atas asset di dalam organisasi.

4. *Update management plan*

Pengendalian atas mutu perangkat lunak ini pun harus melakukan revisi terhadap perencanaan proyek secara keseluruhan, semua stakeholder dapat mengetahui perkembangan dari proyek perangkat lunak secara keseluruhan.



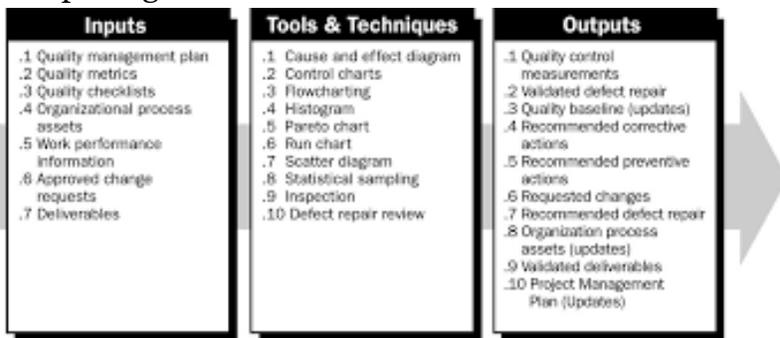
Gambar 10. Kinerja dari pengelolaan mutu proyek

Kendali Mutu (Perform Quality Control)

Pengendalian mutu melibatkan pemantauan dan pengawasan terhadap aktivitas pengembangan perangkat lunak untuk memastikan bahwa standar kualitas terpenuhi. Aktivitas pengendalian ini dilakukan setelah ada produk

yang dihasilkan. Pengendalian mutu dari manajemen proyek perangkat lunak ditujukan untuk meningkatkan mutu dari hasil proyek perangkat lunak yang dijalankan. Keluaran dari kegiatan pengendalian ini yang utama adalah sebagai berikut:

1. **Ketetapan pengakuan**
Ketetapan baik atau jeleknya proyek berlangsung berdasarkan baseline quality ataupun rencana pengendalian mutu
2. **Tindakan korektif dan preventif**
Rekomendasi juga dapat digunakan sebagai informasi bagi tim di dalam melakukan langkah pencegahan atas setiap aktivitas dari proyek yang memberikan kemungkinan terjadinya kesalahan.
3. **Perbaikan proses**
Pimpinan proyek dan tim harus memberikan indikasi dan rekomendasi yang valid atas tingkat kecatatan yang terjadi atau mungkin akan terjadi di dalam proses pengelolaan di dalam proyek perangkat lunak.



Gambar 11. Kerangka Pengawasan mutu dari proyek perangkat lunak

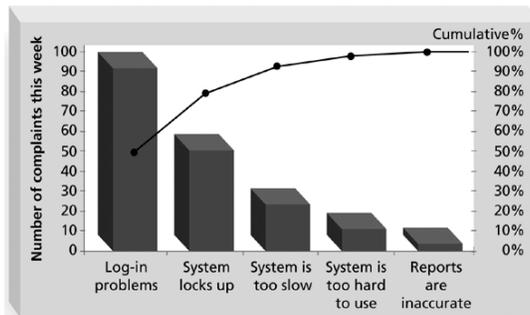
Di dalam prosedur penyusunan manajemen mutu proyek perangkat lunak dapat digunakan beberapa Analisa yang dapat membantu di dalam mengendalikan mutu

proyek. Berikut beberapa peralatan dan Teknik yang bisa digunakan dalam aktivitas pengendalian mutu proyek perangkat lunak:

1. Analisis pareto
2. Pengambilan sampel statistik
3. Six Sigma
4. Grafik control kualitas
5. Testing/pengujian

Analisis Pareto

Analisis pareto adalah salah satu Analisa pengukuran mutu yang dapat digunakan untuk menganalisa masalah yang terkait dengan mutu. Analisa ini menggunakan peraturan 80 – 20, yang berarti 80% masalah yang datang sering disebabkan oleh 20% isu pemicu yang ditimbulkan. Pelaksana proyek harus menyelesaikan masalah yang besar terlebih dahulu kemudian baru ke menyelesaikan masalah yang lebih kecil jika menggunakan analisis pareto. Pareto diagram digambarkan dalam histogram yang membantu proses identifikasi dan prioritas atas area permasalahan.



Gambar 12. Contoh pareto diagram

Pengambilan Sampel Statistik

Perangkat pengukuran Analisa mutu dari proyek

perangkat lunak lainnya dapat diperoleh dengan menggunakan analisa statistik. Analisa statistik ini dilakukan dengan mengukur secara kuantitatif dan berdasarkan suatu tahapan di dalam proyek perangkat lunak yang diambil sebagai contoh dari suatu populasi yang dianalisa. Ukuran dari contoh yang diambil dengan menentukan faktor kepastian dari hasil dan tingkat toleransi penerimaan terhadap mutu yang diharapkan dari hasil proyek. Pengukuran ini sering diformulasikan dalam rumus:

$$\text{Ukuran sampel} = 0,25 \times (\text{Faktor kepastian/kesalahan yang dapat diterima})^2$$

Certainty factor (Faktor kepastian) yang umum digunakan:

Desired Certainty	Certainty Factor
95%	1.960
90%	1.645
80%	1.281

Kepastian 95%: $\text{Ukuran sampel} = 0,25 \times (1,960 / ,05)^2 = 384$

Kepastian 90%: $\text{Ukuran sampel} = 0,25 \times (1,645 / ,10)^2 = 68$

Kepastian 80%: $\text{Ukuran sampel} = 0,25 \times (1,281 / ,20)^2 = 10$

Six Sigma

Analisa pengukuran yang populer di dalam pengukuran mutu yaitu menggunakan metode six sigma. Six sigma merupakan sistem yang adaptif dan lengkap di dalam mengukur mutu suatu produk dengan tujuan untuk memaksimalkan hasil untuk kebutuhan bisnis. Six sigma bersifat unik menggunakan pendekatan yang didasarkan pada kebutuhan pelanggan, penggunaan fakta-fakta, data

dan analisa statistik dengan tujuan untuk memberikan perhatian khusus dalam mengelola, memperbaiki proses bisnis (Pande, Peter S., Robert P. Neuman, and Roland R. Cavanagh, *The Six Sigma Way*, New York: McGraw-Hill, 2000 p.xi).

Target kesempurnaan yang diharapkan untuk dicapai dari penggunaan Six sigma adalah lebih dari 3,4 tingkat kecacatan setiap satu juta kemungkinan dari suatu produk atau proses. Proyek Six Sigma mempunyai 5 tahapan yang sering disebut DMAIC. DMAIC adalah sistematis, proses tertutup untuk perbaikan lanjutan yang berbasis ilmiah dan fakta, DMAIC singkatan yang terdiri atas:

1. Define
Mendefinisikan setiap masalah atau kemungkinan permasalahan terhadap proses kebutuhan pelanggan
2. Measure
Mendefinisikan pengukuran, mengumpulkan, mengolah dan menyajikan data
3. Analysis
Melakukan proses Analisa secara terperinci untuk tujuan memberikan rekomendasi untuk perbaikan
4. Improve
Menyusun suatu langkah penyelesaian dan ide untuk perbaikan
5. Control
Melakuakn pengawasan dan verifikasi atas kestabilan perbaikan serta memberikan prediksi atas penyelesaian masalah dari produksi atau proses

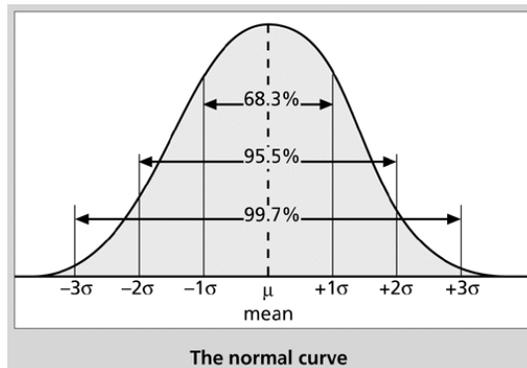
Implementasi dari Six Sigma membutuhkan komitmen yang tinggi dari seluruh personil di dalam organisasi atau perusahaan. Komitmen dari manajemen puncak sangat dibutuhkan untuk keberhasilan implementasi mutu dengan menggunakan Six Sigma.

Perusahaan yang menerapkan Six Sigma mempunyai kemampuan mengurangi tingkat kesalahan, mempercepat proses perbaikan, maupun proses produk yang dihasilkan.

Pendekatan dan fokus pada kebutuhan pelanggan menjadi dasar utama dalam penerapan kebijakan mutu di dalam setiap proyek perangkat lunak. Pengurangan aktivitas yang tidak perlu, peningkatan standar mutu, dan perbaikan kinerja keuangan dengan adanya hasil penerapan proyek perangkat lunak merupakan tujuan utama yang harus dicapai.

Sigma sendiri berarti standar deviasi, yang mengukur selisih dari data dalam bentuk distribusi data. Standar deviasi merupakan faktor kunci di dalam menentukan nilai yang dapat diterima dari tingkat penyimpangan atau cacat yang ditemukan di dalam suatu populasi data. Six Sigma menekankan bahwa tingkat penyimpangan yang diizinkan sebesar 3,4 dari satu juta peluang kejadian atau hasil.

Pengukuran yang sedemikian kecil merupakan standar yang tinggi di dalam setiap penerapan Six Sigma. Distribusi informasi tersebut dapat digambarkan dalam suatu grafik distribusi normal yang disajikan dalam gambar nomor 13 yang menjelaskan sedemikian presisi dari hasil yang ditetapkan di dalam standar penerapan Six Sigma.



Gambar 13. Distribusi normal dan standar deviasi

Tabel 1. Six Sigma dan unit kecacatan

Specification Range (in +/- Sigmas)	Percent of Population Within Range	Defective Units Per Billion
1	68.27	317,300,000
2	95.45	45,400,000
3	99.73	2,700,000
4	99.9937	63,000
5	99.999943	57
6	99.9999998	2

Tabel 2. Konversi six sigma

SIGMA	YIELD	DEFECTS PER MILLION OPPORTUNITIES (DPMO)
1	31.0%	690,000
2	69.2%	308,000
3	93.3%	66,800
4	99.4%	6,210
5	99.97%	230
6	99.99966%	3.4

Keterangan:

Yield: jumlah unit yang dikerjakan dengan tepat melalui prosedurnya.

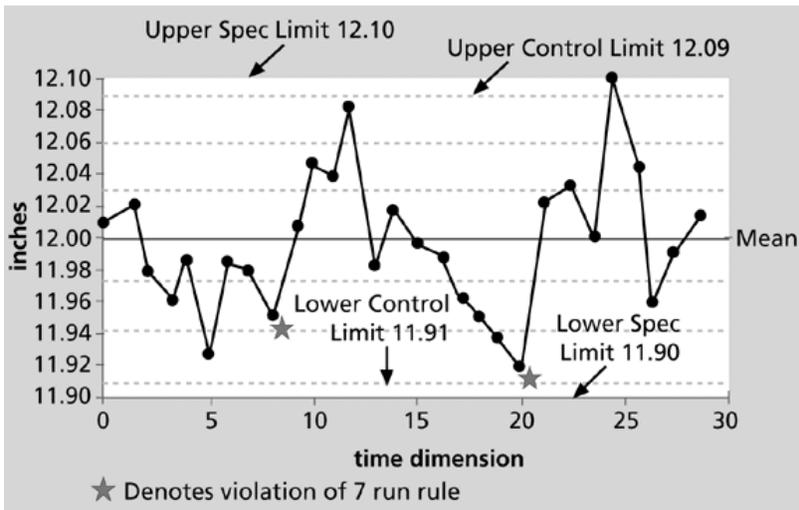
Defect: jumlah unit yang gagal memenuhi kebutuhan pelanggan.

Konvensi Six sigma pada tabel 2 memperhitungkan pergeseran 1,5 sigma untuk diperhitungkan waktu dan mengukur cacat per sejuta peluang, bukan cacat per unit. Makin besar sigma, seharusnya makin banyak unit yang dapat ditangani dengan baik dan bagus mutunya.

Grafik Kontrol Kualitas

Bagan kontrol adalah tampilan grafis dari data yang

mengilustrasikan hasil dari suatu prosedur dari waktu ke waktu. Bagan ini membantu mitigasi resiko dan membantu kita untuk menentukan apakah suatu proses prosedur tersebut terkendali atau tidak. Aturan Seven Run menyatakan bahwa jika 7 (tujuh) titik data dalam satu baris semuanya di bawah rata-rata, di atas rata-rata, atau meningkat atau menurun, maka proses perlu diperiksa untuk masalah non-acak.



Gambar 14. Contoh bagan control mutu

Testing/Pengujian

Para professional teknologi informasi harus memahami dengan baik mengenai standar pemeriksaan dan pengujian aplikasi serta produk perangkat lunak lainnya yang dihasilkan dari sebuah proyek pengembangan perangkat lunak, karena standar pemeriksaan tersebut memberikan suatu ukuran yang jelas di dalam setiap tahapan proyek. Pengujian dan pemeriksaan tidak hanya dilakukan pada tahap akhir pengembangan produk perangkat lunak. Pengujian juga harus dilakukan selama hampir setiap tahapan pengembangan produk perangkat

lunak.

Berikut tipe-tipe pengujian yang bisa dilakukan pada pengembangan produk perangkat lunak:

1. Unit testing

Merupakan pengujian yang dilakukan terhadap setiap komponen program oleh setiap anggota tim dan memastikan bahwa tidak ada tingkat kecacatan yang terjadi pada komponen perangkat lunak yang dibuat

2. Integration testing

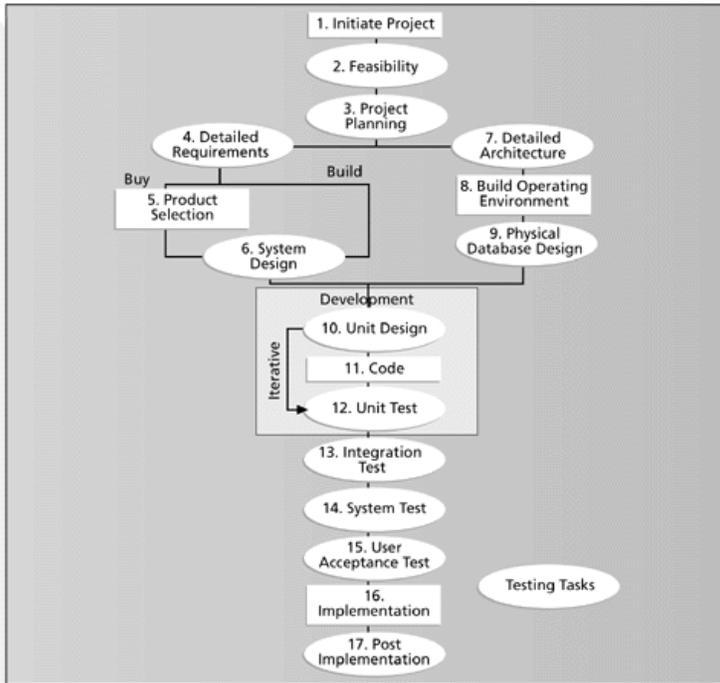
Pengujian ini dilakukan pada tahapan unit dan kombinasi dengan komponen sistem lain secara terintegrasi yang bertujuan untuk melihat sisi fungsional yang dihasilkan.

3. Sistem testing

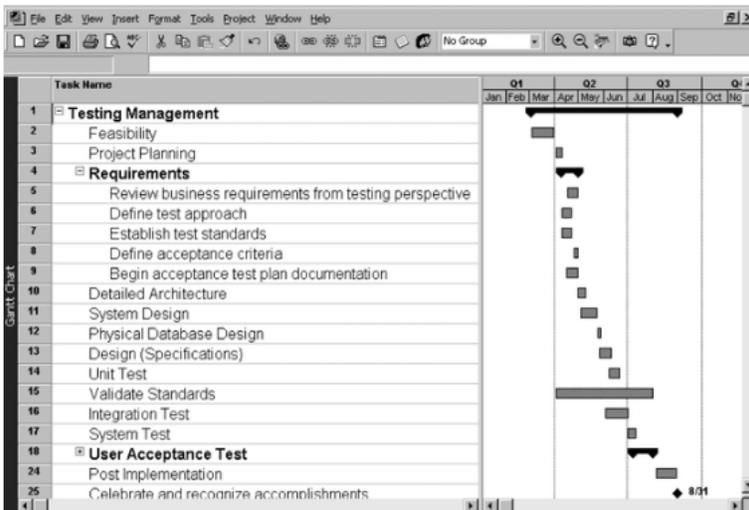
Pengujian ini dilakukan secara keseluruhan dari sistem aplikasi yang ada dan untuk melihat tingkat integritas dari perangkat lunak yang dihasilkan oleh tim proyek.

4. User acceptance testing

Proses pengujian ini dilakukan secara independent oleh pemakai sistem dari perangkat lunak yang dihasilkan oleh tim proyek.



Gambar 15. Pengujian pada tahap SDLC



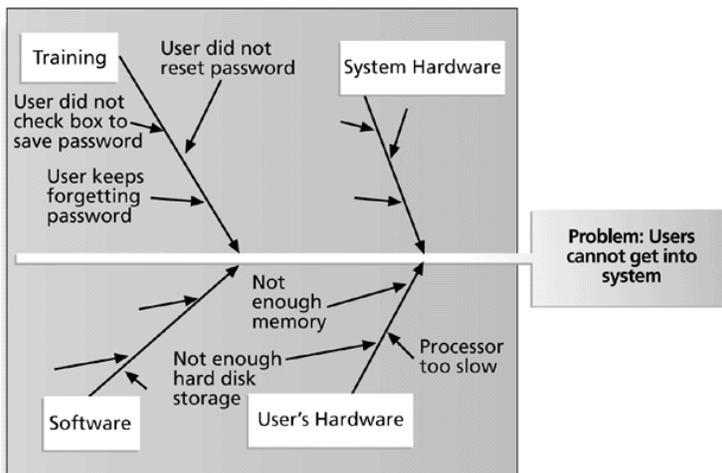
Gambar 16. Gantt Chart untuk Membangun Pengujian dalam System Development Project Plan

Manajemen Mutu Modern

Pendekatan manajemen mutu modern memiliki sejumlah persyaratan yang harus terpenuhi. Persyaratan itu antara lain: tingkat kepuasan pelanggan, mengutamakan pencegahan dibandingkan proses pemeriksaan ulang, menekankan pendekatan tanggungjawab manajemen di dalam keberhasilan atas standar mutu yang ditetapkan.

Berikut daftar beberapa pakar mutu atau kualitas beserta karyanya:

1. Karyanya dalam membangun kembali Jepang dan 14 poinnya membuat Deming menjadi tersohor.
2. Juran menulis Buku Pegangan Kendali Mutu dan 10 langkah peningkatan mutu.
3. Crosby menulis Quality is Free dan menyarankan agar organisasi berjuang untuk tanpa kesalahan.
4. Ishikawa mengembangkan konsep lingkaran kualitas dan memelopori penggunaan diagram Fishbone Taguchi mengembangkan metode untuk mengoptimalkan proses eksperimen Teknik.
5. Feigenbaum mengembangkan konsep kontrol kualitas total.



Gambar 17. contoh fishbone atau diagram ishikawa

Menurut juran hal terpenting di dalam penerapan manajemen mutu pada setiap proyek atau organisasi adalah pola pikir dari manajemen puncak yang memahami dan mengutamakan peningkatan mutu, manifestasi dan kepentingan dari manajemen puncak terhadap mutu walaupun kecil akan berdampak sampai ke jenjang tingkat bawah. Prosentase masalah mutu di dalam proyek yang terbesar sangat berkaitan dengan manajemen puncak dan bukan karena masalah teknis.

Mutu dari proyek perangkat lunak dapat dinaikkan dan diukur berdasarkan beberapa strategi pemahaman yang harus dimiliki oleh pimpinan proyek. Pemahaman tersebut mencakup beberapa tindakan seperti berikut:

1. Kepemimpinan yang dapat meningkatkan dan mengutamakan mutu
2. Memahami biaya dari mutu
3. Focus pada dampak organisasi dan tempat kerja yang mempengaruhi mutu
4. Mengikuti prinsip model kematangan dari sisi teknologi informasi maupun organisasi untuk tujuan menaikkan mutu dari sisi sistem informasi perusahaan.

Biaya Mutu

Manajemen mutu di dalam proyek perangkat lunak akan berdampak terhadap biaya mutu yang harus dikeluarkan untuk tujuan memberikan kontribusi hasil yang layak dan memenuhi standar. Biaya atas mutu (cost of quality) adalah biaya-biaya yang terkait dengan hasil yang harus dicapai dari suatu produk untuk tujuan memenuhi standar kebutuhan dan kelengkapan atas penggunaan produk tersebut. Biaya ini juga mencakup semua tanggungjawab atas kesalahan yang terjadi sebagai akibat

dari mutu yang tidak memenuhi standar atau ketentuan yang telah disepakati. Jadi biaya mutu adalah biaya kesesuaian dan tidak kesesuaian produk hasil dari sebuah proyek.

Table 3. Contoh biaya per jam waktu henti yang disebabkan oleh cacat perangkat lunak

Business	Cost per Hour Downtime
Automated teller machines (medium-sized bank)	\$14,500
Package shipping service	\$28,250
Telephone ticket sales	\$69,000
Catalog sales center	\$90,000
Airline reservation center (small airline)	\$89,500

Biaya mutu dapat dibagi menjadi 5 klasifikasi sebagai berikut:

1. Biaya pencegahan

Biaya atas mutu yang terkait dengan proses perencanaan dan pelaksanaan proyek untuk tujuan meningkatkan mutu dan mengurangi tingkat kesalahan dengan nilai tingkat penerimaan atas mutu hasil yang mendekati sempurna.

2. Biaya penilaian

Biaya atas mutu yang terkait dengan proses evaluasi terhadap proses dan hasil yang dicapai dengan tujuan untuk memastikan tingkat mutu yang memadai.

3. Biaya kegagalan internal

Biaya atas mutu yang terjadi karena proses perbaikan dari suatu kekeliruan yang terdeteksi sebelum pelanggan menerima produk yang dihasilkan dari proyek atau perusahaan

4. Biaya kegagalan eksternal

Biaya atas mutu yang terkait dengan seluruh

kekeliruan yang ditemukan dan perbaikan-perbaikan yang dilakukan sebelum produk yang dihasilkan dikirim atau dipakai oleh pelanggan.

5. Biaya peralatan pengukuran dan pengujian

Biaya atas mutu yang terkait dengan pengadaan investasi peralatan yang dipakai untuk melakukan tindakan preventif dan penilaian setiap aktivitas dari proyek di dalam organisasi.

Sebuah studi oleh DeMarco dan Lister menyatakan bahwa masalah organisasi mempunyai pengaruh yang jauh lebih besar pada produktivitas pemrograman dari pada lingkungan teknis dan Bahasa pemrograman. Produktivitas pemrograman bervariasi dengan faktor satu hingga sepuluh di seluruh organisasi, tetapi hanya sebesar 21% dalam organisasi yang sama. Studi ini tidak menemukan korelasi antara produktivitas dan Bahasa pemrograman, pengalaman bertahun-tahun, atau gaji. Ruang kerja khusus dan lingkungan kerja yang tenang adalah faktor kunci untuk meningkatkan produktivitas pemrograman.

Dengan adanya pengukuran atas biaya mutu tingkat kematangan (maturity models) dari setiap proyek dapat dibuat suatu kategori atau tingkatan yang dapat membantu perusahaan atau proyek perangkat lunak yang dijalankan di dalam meningkatkan mutu dari proses dan sistem. Beberapa model untuk mengukur tingkat kematangan dari pengembangan proyek perangkat lunak meliputi:

1. Model Penyebaran Fungsi Kualitas Perangkat Lunak berfokus pada pendefinisian kebutuhan pengguna dan perencanaan proyek perangkat lunak.
2. Model Kapabilitas Kematangan dari The Software Engineering Institute menyediakan jalur umum untuk meningkatkan proses untuk pengembangan perangkat lunak.
3. Saat ini sekelompok peneliti sedang mengerjakan

model kematangan manajemen proyek, seperti Model Kematangan Manajemen Proyek Organisasi PMI (Organizational Project Management Maturity Model/OPM3).

Model Maturity/Kematangan Manajemen Proyek meliputi:

1. Ad-Hoc

Proses manajemen proyek digambarkan sebagai tidak terorganisir, dan terkadang bahkan kacau. Organisasi belum menetapkan sistem dan proses, dan keberhasilan proyek bergantung pada upaya individu. Ada masalah biaya dan jadwal yang kronis.

2. Abbreviate /Penyingkatan

Ada beberapa proses dan sistem manajemen proyek untuk melacak biaya, jadwal, dan ruang lingkup. Keberhasilan proyek sebagian besar tidak dapat diprediksi dan masalah biaya dan jadwal sering terjadi.

3. Terorganisir

Ada proses dan sistem manajemen proyek yang terstandar dan terdokumentasi yang terintegrasi ke seluruh organisasi. Keberhasilan proyek lebih dapat diprediksi, dan kinerja biaya dan jadwal ditingkatkan.

4. Dikelola

Manajemen mengumpulkan dan menggunakan langkah-langkah rinci dari efektivitas manajemen proyek. Kesuksesan proyek lebih beragam, dan kinerja biaya dan jadwal sesuai dengan rencana.

5. Adaptif

Umpan balik dari proses manajemen proyek dan dari uji coba ide dan teknologi inovatif memungkinkan peningkatan berkelanjutan. Keberhasilan proyek adalah norma, dan kinerja

biaya dan jadwal terus meningkat.

Tingkat Capability Maturity Model (CMM) dari terendah ke tertinggi sebagai berikut:

1. *Repetable*
2. *Defined*
3. *Managed*
4. *Optimizing*

Capability Maturity Model Integration (CMMI) adalah Teknik penerus dari CMM peringkat yang lebih dulu ada untuk mengetahui kematangan sebuah perusahaan atau organisasi dalam pengembangan rekayasa perangkat lunak, rekayasa sistem, dan dalam manajemen program.

BAB 7 MANAJEMEN RESIKO PROYEK PERANGKAT LUNAK

Pendahuluan

Manajemen yang baik melibatkan pengelolaan risiko sebagai elemen integral. Risiko proyek mengacu pada kejadian atau situasi yang tidak dapat diprediksi secara pasti, jika risiko tersebut terjadi dapat memiliki dampak positif atau negatif terhadap tujuan proyek.

Manajemen risiko adalah suatu proses yang terstruktur yang melibatkan perencanaan, identifikasi, analisis, respons, dan monitoring risiko proyek. Ini melibatkan penggunaan proses, alat, serta teknik yang dapat membantu manajer proyek untuk meningkatkan kemungkinan hasil yang positif dan mengurangi kemungkinan serta konsekuensi dari peristiwa yang merugikan yang dapat mempengaruhi tujuan proyek dalam hal biaya, waktu, cakupan, dan kualitas.

Manajemen risiko proyek mencapai efektivitas terbaiknya ketika dilakukan pada tahap awal proyek dan menjadi tanggung jawab yang berkelanjutan sepanjang siklus hidup proyek (Pandian, 2006). Proses manajemen risiko proyek membantu tim proyek dalam mengambil keputusan dengan menggunakan informasi tentang pendekatan alternatif untuk mencapai tujuan serta risiko-risiko yang mungkin muncul dalam proyek tersebut. Manajemen risiko menyediakan cara yang terstruktur untuk mengidentifikasi dan menganalisis potensi risiko, serta merancang dan menerapkan tindakan yang sesuai

dengan dampaknya. Tindakan-tindakan ini umumnya mengacu pada strategi pencegahan risiko, pengalihan risiko, mitigasi dampak atau penerimaan risiko.

Manajemen risiko proyek melibatkan kombinasi ilmu dan seni untuk mengidentifikasi, menganalisis, dan melakukan Tindakan terhadap risiko selama masa proyek, serta memiliki dampak positif dalam pemilihan proyek, penentuan ruang lingkup proyek, dan pengembangan jadwal dan perkiraan biaya yang realistis. Melalui manajemen risiko proyek, para pemangku kepentingan proyek dapat memahami sifat proyek tersebut, melibatkan anggota tim dalam mengidentifikasi kekuatan dan kelemahan, serta membantu mengintegrasikan area pengetahuan manajemen proyek lainnya.

Proses utama dalam manajemen risiko proyek adalah sebagai berikut:

1. Perencanaan manajemen risiko
2. Identifikasi risiko
3. Analisis risiko
 - a. Analisis kualitatif
 - b. Analisis Kuantitatif
4. Perencanaan penanganan risiko
5. Pemantauan risiko

Perencanaan Manajemen Risiko

Rencana manajemen risiko merupakan ringkasan mengenai bagaimana manajemen risiko akan dilaksanakan dalam proyek tertentu. Hal ini penting untuk mengklarifikasi peran dan tanggung jawab, menyusun perkiraan anggaran dan jadwal untuk pekerjaan terkait risiko, serta mengidentifikasi kategori risiko yang perlu dipertimbangkan. Selain itu, penting juga untuk menjelaskan metode pelaksanaan manajemen risiko,

termasuk penilaian probabilitas dan dampak risiko, serta pembuatan dokumentasi terkait risiko. Berikut ini adalah beberapa topik umum yang perlu dibahas dalam rencana manajemen risiko:

1. Metodologi

Manajemen risiko proyek akan dijalankan dengan cara apa? Apa alat dan sumber data yang tersedia dan dapat dimanfaatkan dalam proses manajemen risiko?

2. Peran dan tanggung jawab

Siapa yang bertanggungjawab untuk melaksanakan tugas-tugas tertentu berkaitan dengan manajemen risiko.

3. Anggaran dan jadwal

Berapa perkiraan biaya dan jadwal untuk melakukan aktivitas yang berhubungan dengan risiko?

4. Probabilitas dan Dampak Risiko

Bagaimana penilaian probabilitas dan dampak dari risiko-risiko tersebut akan dilakukan? Metode penilaian dan interpretasi apa yang akan digunakan dalam analisis kualitatif dan kuantitatif risiko?

5. Kategori Risiko

Apa saja kategori risiko utama yang harus ditangani pada proyek ini? Apakah ada struktur perincian risiko untuk proyek.

6. Dokumentasi Risiko

Bagaimana rencana pelaporan akan diterapkan dalam kegiatan manajemen risiko?

Identifikasi Risiko

Tahap identifikasi risiko berkenaan dengan penemuan risiko (Sommerville, 2003) dan mengidentifikasi bahaya yang mungkin terjadi pada suatu proyek. Yang

dikatakan bahaya dalam hal ini adalah suatu keadaan yang dapat terjadi dan menimbulkan masalah terhadap keberhasilan proyek (Ghozali, 2004). Proses Identifikasi Risiko menghasilkan daftar risiko proyek dan secara dokumenter mencatat risiko-risiko yang memiliki potensi mempengaruhi proyek serta mencakup karakteristik risiko tersebut.

Risiko dapat dikelompokkan menjadi beberapa kategori secara umum. Kategori risiko antara lain:

1. Risiko pasar

Apakah produk atau layanan baru yang dihasilkan dari proyek perangkat lunak ini akan memberikan manfaat bagi perusahaan atau dapat dipasarkan kepada pihak yang lain? Apakah pengguna akan menerima layanan atau produk tersebut? Apakah ada kemungkinan bahwa pihak lain akan mengembangkan produk atau layanan yang lebih baik dan lebih cepat, sehingga proyek perangkat lunak ini akan menjadi sia-sia dalam hal waktu dan biaya yang dikeluarkan?

2. Risiko keuangan

Apakah organisasi memiliki kapasitas untuk menjalankan proyek ini? Sejauh mana keyakinan para pemangku kepentingan dalam proyeksi keuangan? Apakah proyek ini akan mencapai Present Value/NPV dan Return on Investment (ROI)? Jika tidak, apakah organisasi mampu melanjutkan proyek ini? Apakah penggunaan keuangan pada proyek ini sudah optimal?

3. Risiko teknologi

Apakah proyek ini memenuhi persyaratan teknis? Apakah proyek ini akan menggunakan teknologi yang telah mapan, terkini, atau yang paling mutakhir? Kapan keputusan akan diambil mengenai

teknologi yang akan digunakan? Apakah semua komponen teknologi, yaitu perangkat keras, perangkat lunak, dan jaringan, akan beroperasi dengan lancar dalam pelaksanaan proyek ini?

4. Risiko sumber daya manusia

Apakah organisasi memiliki atau dapat menemukan individu dengan keterampilan yang sesuai untuk berhasil menyelesaikan proyek ini? Apakah mereka memiliki pengalaman yang memadai? Apakah senior manajemen mendukung proyek ini?

5. Risiko struktur/proses

Seberapa besar dampak yang akan dihasilkan oleh proyek baru ini terhadap area pengguna dan prosedur bisnis? Berapa jumlah kelompok pengguna yang berbeda yang harus terlibat dalam proyek ini? Dan berapa banyak sistem lain yang harus berinteraksi dengan proyek ini? Apakah organisasi memiliki proses yang telah ditetapkan untuk berhasil menyelesaikan proyek ini?

Ada beberapa Teknik dalam indentifikasi risiko.

Antara lain:

1. Interview

Merupakan metode pengumpulan fakta yang melibatkan diskusi tatap muka, telepon, email, atau pesan singkat untuk mengumpulkan informasi. Mewawancarai individu yang memiliki pengalaman dalam proyek serupa dapat menjadi sumber berharga dalam mengidentifikasi potensi risiko.

2. Brainstorming

Tim menggunakan pendekatan ini untuk menghasilkan ide atau menemukan solusi bagi masalah tertentu melalui proses pengumpulan ide secara spontan dan tanpa melakukan penilaian awal. Pendekatan ini dapat membantu membuat daftar

risiko yang komprehensif untuk diatasi dalam proses analisis risiko kualitatif dan kuantitatif. Dalam sesi ini, seorang fasilitator berpengalaman bertugas untuk memimpin dan memfasilitasi agar gagasan-gagasan terus mengalir, sambil memperkenalkan kategori risiko potensial untuk mempertahankan aliran ide. Setelah ide-ide tersebut dikumpulkan, fasilitator dapat mengelompokkan dan mengategorikan ide untuk membuatnya lebih mudah dikelola.

3. Delphi Technique

Teknik ini dimanfaatkan untuk mencapai kesepakatan di antara para ahli yang membuat prediksi tentang perkembangan di masa depan. Teknik ini melibatkan pengumpulan masukan secara independen dan anonim mengenai kejadian-kejadian di masa depan. Proses ini melibatkan pertanyaan dan tanggapan tertulis yang berulang-ulang, menghindari efek bias yang mungkin terjadi dalam metode lisan seperti brainstorming.

4. Analisis SWOT

Analisis SWOT juga dapat digunakan untuk mengidentifikasi risiko dengan mempertimbangkan kekuatan, kelemahan, peluang, dan ancaman yang ada dalam suatu situasi atau proyek dengan cara yang luas, membantu dalam mengenali risiko yang berpotensi berdampak pada proyek.

Analisis Risiko

Proses identifikasi risiko menghasilkan daftar risiko yang mungkin terjadi pada proyek. Berikutnya dilakukan analisis sesuai dengan daftar risiko (Nuris, Maharani, & Rachmadita, 2021). Ada 2 analisis risiko, yaitu:

1. Analisis Kualitatif

Analisis ini menggunakan risk scoring. Risiko yang sudah diidentifikasi dinilai berdasarkan probabilitas terjadinya risiko (likelihood) dan signifikansi atau dampak (consequence).

Probabilitas terjadinya risiko dapat diperkirakan berdasarkan beberapa faktor yang bergantung pada sifat unik dari setiap proyek. Sebagai contoh, faktor-faktor yang dinilai dalam mengidentifikasi risiko potensial dalam teknologi perangkat keras atau perangkat lunak meliputi tingkat kedewasaan teknologi, kompleksitas teknologi, dan ketersediaan dukungan untuk pengembangan teknologi. Dampak dari risiko yang terjadi dapat mencakup konsekuensi seperti ketidakcapaian target kinerja, peningkatan biaya, dan penundaan jadwal.

Dampak yang akan terjadi jika risiko tersebut terjadi dikenal sebagai dampak risiko (risk impact), sedangkan tingkat kepentingan risiko dikenal sebagai nilai risiko (risk value) atau skor risiko (risk score). Nilai risiko dapat dihitung menggunakan formula:

$$\text{Risk value (R)} = \text{risk likelihood (L)} \times \text{risk impact (I)}$$

Contoh tabel risk value (Nuris, Maharani, & Rachmadita, 2021).

Berdasarkan hasil analisis nilai risiko (risk value) dari I1 hingga M7, terdapat risiko-risiko yang memiliki potensi untuk menggagalkan proyek perangkat lunak berdasarkan nilai risiko (R).

2. Analisis Kuantitatif

Dalam analisis berikutnya berdasarkan skor risiko,

dilakukan perhitungan dampak dari setiap risiko menggunakan formula yang telah ditentukan. Kepala proyek menggunakan formula tersebut untuk menentukan faktor risiko, sehingga dapat memudahkan alokasi sumber daya terhadap risiko yang memiliki prioritas.

Analisis risiko kuantitatif melibatkan penerapan teknik statistik, terutama simulasi Monte Carlo yang merupakan salah satu metode yang paling umum dan mudah digunakan, terutama dengan menggunakan perangkat lunak khusus.

Analisis risiko kuantitatif dimulai dengan menggunakan model proyek yang mencakup jadwal proyek dan estimasi biaya, tergantung pada tujuan analisis. Tingkat ketidakpastian dalam setiap aktivitas jadwal dan elemen biaya diwakili oleh distribusi probabilitas. Distribusi probabilitas ini umumnya ditentukan dengan menentukan nilai optimis, nilai yang paling mungkin, dan nilai pesimis untuk setiap aktivitas atau elemen biaya.

Hasil utama dari analisis risiko kuantitatif adalah pembaruan pada daftar risiko, seperti revisi peringkat risiko atau informasi terperinci di balik peringkat tersebut. Analisis kuantitatif juga memberikan informasi tingkat tinggi dalam hal probabilitas pencapaian tujuan proyek tertentu.

Perencanaan Penanganan Risiko

Perencanaan penanganan risiko adalah proses di mana pilihan-pilihan disusun dan tindakan-tindakan ditentukan untuk meningkatkan peluang kesuksesan dan mengurangi ancaman terhadap tujuan proyek. Proses ini berfokus pada daftar risiko yang diidentifikasi sebagai

risiko tinggi dalam analisis risiko kualitatif dan/atau kuantitatif. Dalam perencanaan penanganan risiko, pihak-pihak yang terkait diidentifikasi dan ditugaskan sebagai pemegang tanggung jawab untuk setiap penanganan risiko yang ada. Proses ini memastikan bahwa setiap risiko yang memerlukan penanganan memiliki pemegang tanggung jawab yang ditetapkan.

Manajer dan tim proyek bekerja sama untuk mengidentifikasi strategi terbaik untuk setiap risiko yang ada, dan kemudian merancang tindakan spesifik untuk menerapkan strategi tersebut (Pandian, 2006).

1. Strategi untuk ancaman atau risiko negatif
 - a. Menghilangkan risiko
Proyek direvisi atau disesuaikan dalam rencana untuk menghilangkan risiko atau melindungi sasaran proyek (waktu, cakupan, kualitas) dari dampak risiko yang mungkin terjadi.
 - b. Mentransfer risiko kepada pihak lain
Misalnya melakukan kontrak atau sub pekerjaan dengan pihak lain untuk mengalihkan dampak dari risiko yang mungkin timbul. Transfer risiko ini biasanya akan menimbulkan penambahan biaya.
 - c. Mitigasi atau melakukan Tindakan untuk memperkecil risiko negatif
Melakukan tindakan pencegahan awal untuk mengurangi probabilitas dan dampak risiko seringkali lebih efektif daripada berusaha memperbaiki kerusakan setelah risiko terjadi.
2. Strategi untuk peluang atau risiko positif
 - a. Eksploitasi
Strategi ini bertujuan untuk mengeliminasi ketidakpastian yang terkait dengan risiko

- tertentu dengan mengubah peluang risiko tersebut menjadi pasti terjadi.
- b. Berbagi
Berbagi dengan pihak yang lebih mampu dalam memanfaatkan peluang. Membentuk kemitraan, bekerja sama dengan perusahaan lain.
 - c. Meningkatkan peluang
Strategi ini mengubah skala peluang dengan meningkatkan probabilitas atau dampak yang menguntungkan, serta mengidentifikasi dan memaksimalkan faktor-faktor pendorong utama dari risiko-risiko yang memberikan dampak positif.
 - d. Menerima
Strategi ini dipilih karena tidak memungkinkan untuk menghilangkan risiko dari proyek. Ketika manajer dan tim proyek memutuskan untuk menerima risiko tertentu, mereka tidak perlu melakukan perubahan pada rencana proyek untuk menangani risiko tersebut.

Pemantauan Risiko

Pemantauan adalah kegiatan dalam manajemen risiko yang melibatkan tinjauan dan pengamatan terhadap risiko-risiko yang ada. risiko (Rosmawarni, Ilhamsyah, & Mutiah, 2019). Proses ini juga memantau pelaksanaan strategi yang direncanakan untuk risiko-risiko yang teridentifikasi dan mengevaluasi efektivitasnya.

Pemantauan dan pengendalian risiko terus dilakukan sepanjang masa proyek. Daftar risiko proyek akan mengalami perubahan seiring dengan kemajuan

proyek, dengan munculnya risiko baru, dan dengan hilangnya risiko yang telah diantisipasi sebelumnya.

Penting untuk mengadakan pertemuan guna membahas risiko yang diadakan secara teratur, dimana semua atau sebagian dari daftar risiko ditinjau untuk efektivitas penanganannya dan risiko baru didiskusikan serta penanggungjawabnya. Tinjauan risiko proyek secara berkala mengulangi proses identifikasi, analisis, dan perencanaan penanganan. Manajer proyek memastikan bahwa risiko proyek menjadi agenda dalam semua pertemuan tim. Peringkat dan prioritas risiko biasanya berubah selama siklus hidup proyek.

Apabila muncul risiko yang tidak terantisipasi atau dampak risiko yang lebih besar dari yang diperkirakan, mungkin diperlukan perencanaan penanganan tambahan agar risiko dapat dikendalikan dengan baik. Manajer dan tim proyek harus melakukan perencanaan penanganan tambahan untuk mengatasi risiko tersebut.

BAB 8 DOKUMEN KONTRAK

Pendahuluan

Mengenai Sistem kontrak sebuah tujuan untuk mengikat 1 atau 2 atau 3 lebih orang untuk melakukan hal kesepakatan tertentu dengan menitik beratkan tujuan bersama dan saling menguntungkan berbagai pihak, serta dalam dokumen kontak pula dapat memberikan rambu rambu antar kepentingan agar kedepannya salah satu pihak tidak mengalami kerugian baik moral serta materil, jika ditarik berdasarkan hukum yang berlaku untuk dianggap sah berdasarkan hukum ada 4 poin utama yaitu: adanya kesepakatan dari yang berkepentingan, profesionalitas kerja, hal hal yang penting dijabarkan di surat, sebab akibat diperbolehkan hukum.

Jika adanya pembatalan pembatalan yang dimuat di kesepakatan yang tertera dapat dibatalkan atau diteruskan ke jalur hukum atau dimusyawarahkan terlebih dahulu, jika pada kesepakatan awal ada hal hal tertentu yang dirasa tidak fair atau merugikan salah satu pihak, misal adanya penambahan dana, proyek yang tidak sesuai, keterlambatan proyek, tidak keseuaian proyek awal yang diminta serta hal hal yang diluar dugaan lainnya. Dokumen kontrak harus diberikan saat awal kesepakatan proyek, dengan adanya dokumen kontrak sistem pekerjaan kita akan semakin tertata baik waktu, manajemen tim serta hasil sistem informasi yang dibuat akan menjadi efisien dan handal saat digunakan trial maupun saat post proyek.

Surat Perjanjian

Dalam melakukan surat perjanjian kerjasama ada 2 orang yang melakukan perjanjian kerja, yang dimana pihak pertama sebagai pihak utama melakukan perjanjian kerja dengan pihak kedua dengan menyertakan biodata lengkap kedua belah pihak, selanjutnya pasal pasal yang mengikat keduanya dijabarkan kedalam surat perjanjian kerja misal:

Ayat 1

PIHAK PERTAMA menyatakan menerima PIHAK KEDUA sebagai pemegang projek teknologi informasi (empat Tahun) di kampus/perusahaan..... yang beralamat di Jl. dan PIHAK KEDUA dengan ini menyatakan kesediaannya.

Ayat 2

Perjanjian kerja ini berlaku untuk jangka waktu Tahun, terhitung sejak tanggal dan berakhir pada tanggal

Ada 3 aspek utama dalam melakukan perjanjian kerja, yang harus dipenuhi saat mengerjakan sebuah projek yaitu

1. Perbuatan

Isi segala bentuk tujuan dari surat perjanjian kerja itu dibuat

2. Pelaku

Dua belah pihak yang saling mengikat perjanjian kerja yang saling berkepentingan

3. Pengikat

Pihak pihak siapapun terlibat diperjanjian sama dimata hukum jadi konsekuensi konsekuensi harus dibahas juga di perjanjian kerja tersebut

Terdapat fungsi-fungsi juga dari surat perjanjian

kerja ini yaitu:

1. Sebagai dasar kesepakatan kedua belah pihak yang melakukan perjanjian kerja
2. Sebagai pedoman gugatan jika salah satu pihak melakukan pemangkiran tanggung jawab yang tertuang di perjanjian kerja.
3. Sebagai alat yang menghindarkan dari persengketaan atau hal yang diluar dari perjanjian kerja.

Surat Perjanjian Rangka Penunjukan

Istilah-istilah yang digunakan dalam Syarat-Syarat Umum Kontrak ini harus mempunyai arti atau tafsiran seperti yang dimaksudkan sebagai berikut:

1. Aplikasi yang akan di tenderkan adalah setiap aplikasi berbasis web atau mobile, yang nantinya akan digunakan oleh calon user baik untuk perbaikan manajemen dan seterusnya.
2. Pengguna anggaran yaitu CV atau PT yang ditunjuk untuk mengerjakan sistem informasi manajemen tersebut.
3. Tim Pembuat Komitmen yang bertanggung jawab atas pelaksanaan pengadaan barang misal untuk pembelian print, server, komputer dan kabel jaringan dan seterusnya.
4. Tim pengawas Internal sebagai orang yang melakukan komunikasi dengan tim IT untuk kegiatan flowchart, evaluasi serta testing program.
5. Penyedia pihak ketiga biasanya untuk penyedia barang penunjang sistem.
6. Surat penjaminan yaitu surat progress IT dipastikan program bisa berjalan dengan baik dan sesuai deadline.

7. Kontrak Pengadaan Barang/Jasa.
8. Nilai kontrak pembuatan program adalah total harga pembuatan program yang tercantum dalam kontrak.
9. Time table pengerjaan sistem informasi manajemen.
10. Daftar kuantitas dan harga (rincian harga penawaran) adalah daftar kuantitas yang telah diisi harga satuan dan jumlah biaya keseluruhannya yang merupakan bagian dari penawaran.
11. Masa Kontrak proyek, waktu berlakunya kontrak ini terhitung sejak tanggal penandatanganan kontrak antara pihak pertama yaitu klien dan kedua yaitu tim IT kita sampai dengan serah terima program aplikasi.

Langkah-Langkah Analisa Project yang akan Dimasukan ke Proposal

1. Analisa kebutuhan *software*
Perancangan flowchart dengan baik akan menciptakan alur sistem yang baik mulai dari login dan masuk menu utama, dengan adanya flowchart secara baik membantu tim IT dan Client dalam menemukan pola sistem informasi manajemen yang terukur, efisien dan sesuai dengan terapan pekerjaan yang sudah di kerjakan sebelumnya.
2. Desain UI/UX
Penggambaran *user interface front end* yang baik akan menciptakan kesan kepada clien bawasannya sistem yang akan dibuat akan terwujud seperti yang mereka inginkan, serta mempermudah client dalam memahami desain form yang mereka berikan ke kita bisa dideskripsikan dengan baik, sesuai dengan permintaan mereka. Desain UI/UX ini akan memakan banyak data berkaitan dengan itu karena

akan bermuara ke flowchart sistem dan ujung ujungnya akan ke pengaturan database sistem.

3. Database

Untuk merancang aplikasi software berbasis infrastruktur sehingga menggambarkan antar relasi dari suatu data. Dengan perancangan database yang baik diharapkan antar relasi dapat berjalan dengan baik.

4. Code generation

Dalam merancang sistem informasi manajemen tidak lepas akan sistem keamanan data yang akan diterapkan kedalam program, selain itu juga merancang bagaimana sistem nanti bisa dikontrol penuh oleh atasan hingga tidak ada kecurangan data dalam penginputan bahkan proses lainnya. Serta mempermudah juga audit data dari tim internal mereka.

5. Testing

Proses dimana semua divisi didalam manajemen klien melakukan proses input output dst, untuk memastikan sistem berjalan dengan sesuai yang diminta saat awal kontrak berjalan, serta tim IT memastikan tidak ada problem sistem saat testing berjalan biasanya testing ini dilakukan paling lama 1 bulan sebelum serah terima dilaksanakan. Pastikan juga saat testing pembahasannya tidak melenceng terhadap dokumen kontrak yang berjalan, jika ada tim internal dirasa melenceng harus di putusakan ulang surat penawaran yang baru agar kita sebagai tim IT tidak terkuras akan hal yang bukan dalam koridor kesepakatan awal.

Surat Penawaran

Dalam merancang surat penawaran ada beberapa hal yang harus di tentukan sebelum diberikan ke *client* sehingga nantinya clien bisa mengkalkulasi kemampuan buget yang akan dikeluarkan tanpa mengurangi performa siste informasi manajemen itu sendiri dan tanpa merugikan tim IT yang membuat serta merancangnya yaitu:

1. Informasi biaya pembuatan UI (*user interface* antarmuka) sistem yang diinginkan karena membangun sebuah user interface yang ideal butuh tim sendiri untuk memproyeksikan
2. Informasi manajemen sistem mulai dari pembuatan halaman admin, halaman user, manajemen postingan, manajemen member, report keuangan, statistic data laporan yang kompleks dst menyesuaikan permintaan dari klien.
3. Perangkat hosting yang akan digunakan besar atau kecilnya disesuaikan dengan kebutuhan dari klien.
4. Proses ujicoba atau testing, dalam hal ini akan ada penunjukan perwakilan tiap divisi pada internal perusahaan atau lembaga untuk melakukan testing atau ujicoba dari data yang sebelumnya diberikan kepada kita

Tidak bisa dipungkiri surat penawaran akan selalu berubah-berubah seiring dengan kesepakatan kedua bela pihak, ada beberapa faktor yang mempengaruhi perubahan perubahan yang terjadi di surat penawaran:

1. Client memiliki keinginan untuk bekerja sama secara berkelanjutan biasanya perusahaan baru dan belum memiliki tim IT yang handal akan melakukan kerjasama berkelanjutan ini.
2. Client memiliki budget keuangan yang standart tapi ingin memakai jasa kita.

3. Client memiliki masalah yang kompleks dengan pekerjaannya atau integrasi data yang kompleks.
2. Client pasrah projek ke kita dengan berbagai catatan.
3. Client memiliki data yang acak acakan yang banyak, saat ingin membuat aplikasi.

Berikut tabel adalah contoh dari dokumen penawaran pada cv ideutama:

Tabel 4. Tabel penawaran

Type	Keterangan	Harga	Waktu
Membuat layout antar muka	Menu – menu yang disediakan sbb : Design Layout Dashboard Slideshow Postingan	Rp 2.000.000	7 hari
Manajemen sistem	Menu – menu yang disediakan sbb : Halaman admin Halaman User Manajemen posting Manajemen member Report keuangan Statistik data	Rp 4.000.000	14 hari
Hosting	Menu – menu yang disediakan sbb : 1 GB hosting Setting hosting	Rp 1.500.000	1 hari
Ujicoba	<ol style="list-style-type: none"> 1. Ujicoba untuk menghindari error pada aplikasi 2. Pelatihan dan pendampingan untuk 	Free	4 hari

	karyawan yang akan menggunakan aplikasi		
	Total	Rp 7.500.000	26 hari
<i>Adapun harga – harga tersebut belum termasuk PPN 11%</i>			

Surat Pengadaan Barang Penunjang

Aplikasi

Dalam merancang aplikasi yang kompleks yang akan digunakan oleh tim perusahaan atau lembaga yang besar atau pun kecil, pengadaan barang jadi penting disisipkan karena berbeda dengan surat penawaran sistem, surat pengadaan barang berfokus pada barang-barang yang nantinya digunakan untuk menunjang berjalannya aplikasi. Banyak factor yang mempengaruhi surat pengadaan barang disetujui atau tidak dalam dokumen kontrak yaitu:

1. Merk barang yang tidak sesuai dengan kebutuhan client, memang ada klien yang memiliki fanatik merk tertentu atau tidak.
2. Harga yang kurang kompetitif yang dirasakan client atau terkesan mahal karena punya refrensi merk lain
3. Client belum membutuhkan pengadaan barang hanya aplikasi saja yang dibutuhkan karena perusahaan nya masih skala kecil.
2. Pengadaan barang yang terkesan memaksakan klien.

Oleh karena itu pastikan surat pengadaan barang ini digunakan dengan semestinya, jangan terkesan diadadakan atau memaksa clien untuk membeli, ikuti kemauan klien bagaimana nanti kita sesuaikan budget klien minta. Pentingnya surat pengadaan barang untuk perusahaan

klien yaitu:

1. Sebagai bukti yang jelas antara kedua belah pihak
2. Sebagai dokumen pertanggungjawaban
3. Sebagai juga arsip perusahaan
4. Memudahkan untuk saat perpindahan barang di perusahaan

Surat Perintah Kerja

Dalam proses pengerjaan proyek pembuatan aplikasi hal-hal yang banyak dibicarakan adalah kapan kita mulai mengerjakan setelah penawaran dan kontrak sudah di dapatkan, surat perintah kerja memiliki hal yang mengikat karena berkaitan jadwal penyelesaian yang harus dikerjakan dengan tepat waktu berikut fungsi surat perintah kerja:

1. Memberikan perlindungan kepada pemenang tender.
2. Sebagai dokumen yang sah untuk mengambil data pada tiap divisi perusahaan atau lembaga.
3. Sebagai ijin yang tertulis keluar masuk lembaga atau perusahaan selama masa pengerjaan program.
4. Sebagai informasi kepada pihak pihak tertentu atas proyek yang dikerjakan.

Untuk bagian bagian surat perintah kerja terdiri dari beberapa komponen yaitu:

1. Kop surat
2. Nomer surat
3. Identitas yang menerima surat perintah kerja
4. Identitas yang memberikan surat perintah kerja
5. Isi surat perintah kerja
6. Rincian surat perintah
7. Nama terang dan tanda tangan yang bersangkutan atas surat perintah kerja

BAB 9 ORGANISASI DAN PERSONEL PROYEK PERANGKAT LUNAK

Pendahuluan

Organisasi dan personel yang efektif merupakan elemen kunci dalam kesuksesan manajemen proyek perangkat lunak. Dalam Bab ini, akan dibahas pentingnya struktur organisasi yang tepat dan peran personel yang terdefinisi dengan baik dalam mencapai tujuan proyek perangkat lunak.

Struktur organisasi proyek perangkat lunak mencakup bagaimana tanggung jawab, wewenang, dan aliran komunikasi diatur dalam tim. Struktur organisasi yang baik memungkinkan koordinasi yang efektif, pengambilan keputusan yang cepat, dan pencapaian tujuan proyek.

Personel dalam proyek perangkat lunak adalah anggota tim yang terlibat langsung dalam perancangan, pengembangan, pengujian, dan implementasi perangkat lunak. Setiap anggota tim memiliki peran dan tanggung jawab yang berbeda sesuai dengan keahlian dan spesialisasi mereka.

Dengan struktur organisasi yang tepat dan personel yang terampil, proyek perangkat lunak memiliki pondasi yang kokoh untuk mencapai tujuan yang ditetapkan. Melalui pendekatan yang baik terhadap organisasi dan personel, proyek perangkat lunak dapat dilaksanakan dengan efisien, berkualitas, dan tingkat kesuksesan yang tinggi.

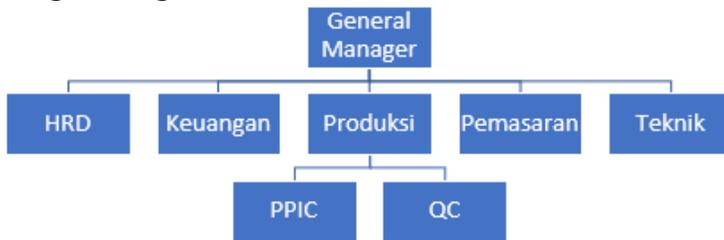
Struktur Organisasi Proyek Perangkat Lunak

Lunak

Struktur Organisasi Proyek perangkat lunak merujuk pada cara tim dan peran terorganisir dalam proyek tersebut. Struktur organisasi proyek yang baik memastikan bahwa tanggung jawab, wewenang, dan alur komunikasi yang jelas ada di antara anggota tim proyek. Berikut adalah beberapa struktur organisasi yang umum digunakan dalam proyek perangkat lunak:

1. Struktur Organisasi Fungsional

Struktur Organisasi Fungsional adalah jenis struktur organisasi yang umum digunakan di berbagai organisasi. Dalam struktur ini, pekerjaan dan orang-orang dikelompokkan berdasarkan fungsi atau spesialisasi tertentu. Setiap fungsi memiliki departemen atau bagian sendiri yang bertanggung jawab atas tugas dan tanggung jawab yang terkait dengan fungsi tersebut.



Gambar 18. Organisasi Fungsional

Keuntungan dari Struktur Organisasi Fungsional:

a. Pemisahan Fungsi

Struktur Organisasi Fungsional memisahkan fungsi atau spesialisasi tertentu dalam kelompok yang terpisah. Misalnya, ada departemen produksi, keuangan, pemasaran, dan seterusnya. Ini memungkinkan orang-

- orang dengan keterampilan dan pengetahuan yang sama untuk bekerja bersama dalam departemen mereka masing-masing.
- b. Keahlian dan Spesialisasi
Dalam struktur ini, individu cenderung mengembangkan keahlian dan spesialisasi dalam fungsi atau departemen mereka. Ini memungkinkan tim yang terampil dan berpengetahuan dalam area tertentu untuk mengatasi tugas dan tantangan yang spesifik dalam fungsi mereka.
 - c. Pengambilan Keputusan yang Konsisten
Dalam Struktur Organisasi Fungsional, keputusan yang berkaitan dengan fungsi atau departemen diambil oleh manajer atau pemimpin departemen tersebut. Hal ini dapat menghasilkan keputusan yang konsisten dan sesuai dengan pengetahuan dan pengalaman yang dimiliki oleh departemen tersebut.
 - d. Penggunaan Sumber Daya yang Efisien
Dalam struktur ini, sumber daya seperti peralatan, personel, dan keuangan dapat digunakan secara efisien. Setiap departemen bertanggung jawab atas sumber daya yang terkait dengan fungsi mereka, dan mereka dapat mengelolanya sesuai dengan kebutuhan mereka.
 - e. Pengembangan Karier yang Jelas
Struktur Organisasi Fungsional sering menawarkan jalur karier yang jelas dalam fungsi atau departemen tertentu. Individu dapat mengembangkan keterampilan dan pengetahuan mereka dalam bidang yang mereka minati dan bergerak maju dalam

peringkat atau posisi yang lebih tinggi dalam departemen tersebut.

f. Peningkatan Koordinasi Intern

Dalam struktur ini, koordinasi antara departemen yang berbeda biasanya terjadi melalui hierarki manajemen yang ditetapkan. Setiap departemen bekerja sama dengan departemen lain untuk mencapai tujuan organisasi secara keseluruhan.

Namun, Struktur Organisasi Fungsional juga memiliki beberapa keterbatasan, seperti:

a. Kurangnya Komunikasi Lintas Fungsi

Terkadang, komunikasi antara departemen atau fungsi yang berbeda dalam Struktur Organisasi Fungsional menjadi sulit. Hal ini dapat menghambat kolaborasi dan koordinasi antara departemen.

b. Kurangnya Fleksibilitas

Karena fokus pada fungsi atau departemen tertentu, struktur ini mungkin kurang fleksibel dalam menangani perubahan dan tantangan yang muncul di luar batas fungsi tersebut.

c. Kurangnya Fokus pada Proyek atau Tujuan Khusus

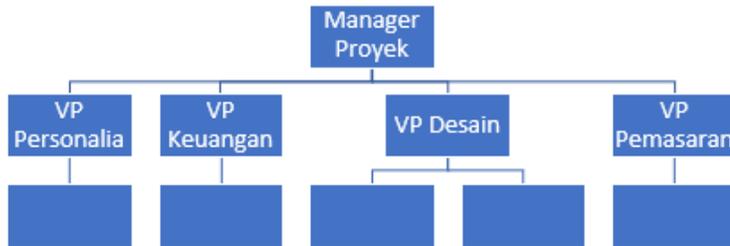
Struktur Organisasi Fungsional mungkin kurang efektif dalam mengelola proyek atau tujuan khusus yang membutuhkan kerjasama lintas fungsi. Fokus yang terlalu kuat pada departemen dapat menghambat sinergi dan kolaborasi antara tim.

Struktur Organisasi Fungsional dapat menjadi pilihan yang baik untuk organisasi yang beroperasi dalam lingkungan yang stabil, dengan fokus pada

keahlian dan spesialisasi dalam fungsi tertentu. Namun, perlu dipertimbangkan keadaan dan kebutuhan organisasi sebelum memilih struktur organisasi yang tepat.

2. Struktur Organisasi Proyek murni

Struktur Organisasi Proyek Murni adalah bentuk struktur organisasi yang dikhususkan untuk mengelola proyek-proyek tertentu. Dalam struktur ini, tim proyek dibentuk secara independen dan secara eksklusif bekerja pada proyek tertentu. Tim proyek memiliki otonomi penuh dalam pengambilan keputusan dan pengelolaan sumber daya yang terkait dengan proyek mereka. Setiap proyek memiliki manajer proyek yang bertanggung jawab atas keberhasilan proyek tersebut.



Gambar 19. Organisasi Proyek Murni

Karakteristik Struktur Organisasi Proyek Murni

a. Fokus pada Proyek

Struktur ini dirancang khusus untuk memusatkan perhatian dan sumber daya organisasi pada proyek-proyek tertentu. Setiap proyek memiliki tim yang ditugaskan secara khusus untuk mengerjakan proyek dengan tujuan mencapai hasil proyek yang diinginkan.

b. Tim Proyek Mandiri

Dalam Struktur Organisasi Proyek Murni,

setiap tim proyek bekerja secara mandiri dan memiliki otonomi dalam mengelola proyek. Mereka memiliki wewenang penuh dalam pengambilan keputusan dan penggunaan sumber daya yang relevan dengan proyek mereka.

c. Manajer Proyek yang Kuat

Setiap proyek dalam Struktur Organisasi Proyek Murni memiliki seorang Manajer Proyek yang ditunjuk. Manajer Proyek memiliki tanggung jawab penuh atas keberhasilan proyek tersebut, termasuk pengelolaan anggaran, jadwal, sumber daya, dan pengambilan keputusan terkait proyek.

d. Sumber Daya yang Dikhususkan

Dalam struktur ini, sumber daya organisasi dialokasikan secara khusus untuk proyek-proyek tertentu. Anggota tim proyek, termasuk personel, peralatan, dan fasilitas, didedikasikan sepenuhnya untuk proyek tersebut selama periode proyek.

e. Komunikasi Terfokus

Komunikasi dalam Struktur Organisasi Proyek Murni cenderung lebih terfokus dan langsung. Tim proyek berkomunikasi secara efektif di antara anggota tim dan dengan Manajer Proyek, kerjasama yang lebih baik dan pemecahan masalah yang lebih cepat.

Keuntungan Struktur Organisasi Proyek Murni meliputi:

- a. Fokus pada tujuan dan keberhasilan proyek.
- b. Tim yang terdedikasi sepenuhnya untuk proyek, meningkatkan keterampilan dan pengetahuan khusus.

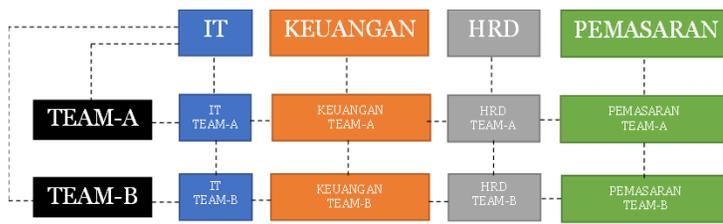
- c. Pengambilan keputusan yang cepat dan efisien karena tim proyek memiliki wewenang penuh.
- d. Komunikasi yang terfokus dan efektif antara anggota tim proyek.
- e. Fleksibilitas dalam menyesuaikan sumber daya dan prioritas sesuai dengan kebutuhan proyek.

Namun, Struktur Organisasi Proyek Murni juga memiliki beberapa keterbatasan. Misalnya:

- a. Dalam situasi ketika proyek tidak membutuhkan sumber daya yang sepanjang waktu, struktur ini mungkin tidak efisien dalam penggunaan sumber daya.
- b. Terdapat potensi terjadinya duplikasi sumber daya dan upaya antara tim proyek, yang dapat mengakibatkan pemborosan dan ketidakefisienan.
- c. Koordinasi antara tim proyek yang berbeda dalam organisasi bisa menjadi lebih rumit, karena masing-masing tim bekerja secara independen.

3. Struktur Organisasi Matrik

Struktur Organisasi Matrik adalah jenis struktur organisasi yang menggabungkan elemen-elemen struktur organisasi fungsional dan struktur organisasi proyek. Dalam struktur ini, anggota tim proyek bekerja di bawah kepemimpinan Manajer Proyek, tetapi juga tetap memiliki ikatan dengan departemen fungsional mereka. Dalam Struktur Organisasi Matrik, ada dua garis wewenang yang ada, yaitu garis wewenang fungsional dan garis wewenang proyek.



Gambar 20. Organisasi Matrik

Karakteristik Struktur Organisasi Matrik

a. Dual Reporting

Dalam Struktur Organisasi Matrik, anggota tim proyek memiliki dua atasan. Mereka melapor kepada Manajer Proyek untuk hal-hal yang terkait dengan proyek dan melapor kepada Manajer Fungsional atau Departemen untuk masalah-masalah yang terkait dengan pekerjaan rutin atau fungsional mereka.

b. Fokus pada Proyek dan Fungsi

Struktur Organisasi Matrik menggabungkan fokus pada proyek dengan keahlian dan spesialisasi fungsional. Tim proyek yang terbentuk berdasarkan proyek memiliki tanggung jawab khusus dalam mencapai tujuan proyek, sementara fungsi fungsional tetap bertanggung jawab atas pekerjaan rutin dan dukungan fungsional.

c. Kolaborasi dan Koordinasi

Struktur Organisasi Matrik mendorong kolaborasi dan koordinasi antara anggota tim proyek dan departemen fungsional. Komunikasi dan kerjasama yang baik diperlukan untuk memastikan bahwa sasaran proyek terpenuhi sambil memenuhi kebutuhan fungsional organisasi.

- d. Fleksibilitas dalam Penggunaan Sumber Daya
Dalam Struktur Organisasi Matrik, sumber daya dapat dialokasikan secara fleksibel dan dapat digunakan oleh beberapa proyek sekaligus. Hal ini memungkinkan penggunaan yang lebih efisien dari sumber daya organisasi yang terbatas.
- e. Keahlian Khusus
Struktur Organisasi Matrik memungkinkan anggota tim proyek untuk mengembangkan keahlian khusus dalam proyek tertentu, sementara juga mempertahankan keahlian dan keterampilan fungsional mereka. Hal ini dapat meningkatkan kualitas dan efisiensi pekerjaan yang dilakukan dalam proyek.

Keuntungan dari Struktur Organisasi Matrik meliputi:

- a. Kombinasi keahlian fungsional dan fokus pada proyek.
- b. Fleksibilitas dalam penggunaan sumber daya organisasi.
- c. Kemampuan untuk mengoptimalkan penggunaan sumber daya manusia yang terbatas.
- d. Peningkatan komunikasi dan koordinasi antara tim proyek dan departemen fungsional.

Namun, Struktur Organisasi Matrik juga memiliki beberapa kekurangan, seperti:

- a. Potensi konflik kepentingan antara Manajer Proyek dan Manajer Fungsional dalam mengalokasikan sumber daya.
- b. Kompleksitas dalam mengelola komunikasi dan koordinasi di antara anggota tim proyek dan departemen fungsional.

- c. Kesulitan dalam mengembangkan mekanisme pengambilan keputusan yang jelas dan proses manajemen konflik yang efektif.

Personel Proyek Perangkat Lunak

Personel Proyek Perangkat Lunak merujuk kepada anggota tim yang terlibat dalam pengembangan perangkat lunak. Setiap anggota tim memiliki peran dan tanggung jawab yang spesifik dalam mencapai tujuan proyek. Berikut ini adalah beberapa peran umum dalam personel proyek perangkat lunak:

1. Manajer Proyek
 - a. Bertanggung jawab atas perencanaan, pengorganisasian, pengendalian, dan pengawasan keseluruhan proyek perangkat lunak.
 - b. Menetapkan tujuan proyek, mengelola resiko, mengawasi anggaran dan jadwal, serta berkomunikasi dengan pihak terkait eksternal.
2. Analis Sistem
 - a. Bertanggung jawab untuk mengumpulkan, menganalisis, dan memahami kebutuhan pengguna atau klien.
 - b. Menerjemahkan kebutuhan pengguna menjadi persyaratan fungsional yang dapat dimengerti oleh tim pengembang.
3. Perancang Sistem
 - a. Merancang arsitektur sistem secara keseluruhan, termasuk struktur komponen, interaksi, dan infrastruktur teknologi yang akan digunakan.
 - b. Memastikan bahwa sistem yang dirancang

- memenuhi persyaratan dan standar kualitas yang ditetapkan.
4. Pengembang Perangkat Lunak
 - a. Menulis kode program untuk mengimplementasikan fungsi-fungsi perangkat lunak sesuai dengan desain dan persyaratan yang telah ditentukan.
 - b. Melakukan pengujian unit dan debugging untuk memastikan kehandalan dan kualitas perangkat lunak.
 5. Pengujian Perangkat Lunak
 - a. Merancang strategi pengujian untuk memverifikasi dan validasi perangkat lunak.
 - b. Melakukan pengujian fungsional, pengujian integrasi, pengujian performa, dan pengujian lainnya untuk memastikan kualitas perangkat lunak sebelum diluncurkan.
 6. Manajer Konfigurasi
 - a. Bertanggung jawab untuk mengelola konfigurasi perangkat lunak, termasuk versi perangkat lunak, kontrol perubahan, dokumentasi, dan manajemen konfigurasi.
 7. Dokumentator
 - a. Bertanggung jawab untuk membuat dan memelihara dokumentasi proyek, termasuk dokumentasi teknis, panduan pengguna, dokumentasi desain, dan dokumentasi lainnya.
 8. UI/UX Designer
 - a. Merancang antarmuka pengguna yang intuitif, menarik, dan mudah digunakan.
 - b. Menggunakan prinsip-prinsip desain UX untuk memastikan pengalaman pengguna yang baik.

9. Manajer Kualitas

- a. Bertanggung jawab memastikan bahwa perangkat lunak memenuhi standar kualitas yang ditetapkan.
- b. Mengembangkan dan menerapkan strategi pengujian kualitas serta memantau kesesuaian proyek dengan proses pengembangan yang ditetapkan.

10. Koordinator Proyek

- a. Memfasilitasi komunikasi antara anggota tim, memastikan koordinasi tugas, dan mengelola jadwal proyek.
- b. Melakukan pemantauan kemajuan proyek, mengidentifikasi dan mengatasi hambatan, serta memastikan bahwa proyek berjalan sesuai dengan jadwal yang ditetapkan.

Setiap anggota tim memiliki peran penting dalam mencapai kesuksesan proyek pengembangan perangkat lunak. Penting untuk memastikan bahwa setiap peran memiliki deskripsi tugas yang jelas, komunikasi yang efektif, dan kolaborasi yang baik dalam tim.

Kepemimpinan Proyek Perangkat Lunak

Perilaku kepemimpinan dalam proyek perangkat lunak berperan penting dalam mengarahkan tim, memotivasi anggota tim, dan mencapai tujuan proyek. Berikut ini beberapa perilaku kepemimpinan yang relevan dalam konteks proyek perangkat lunak:

1. Kepemimpinan Berorientasi pada Visi

Seorang pemimpin proyek perangkat lunak yang efektif memiliki visi yang jelas tentang tujuan proyek dan kemampuan untuk mengkomunikasikan visi tersebut kepada tim. Mereka mampu menginspirasi

dan menggerakkan anggota tim dengan menggambarkan gambaran masa depan yang memotivasi.

2. Keterampilan Komunikasi yang Kuat

Seorang pemimpin proyek perangkat lunak harus memiliki keterampilan komunikasi yang baik untuk berinteraksi dengan anggota tim, pemangku kepentingan, dan pihak terkait lainnya. Mereka harus dapat mengungkapkan gagasan dengan jelas, mendengarkan dengan empati, dan memfasilitasi diskusi yang efektif.

3. Delegasi yang Efektif

Seorang pemimpin proyek perangkat lunak harus mampu mendistribusikan tugas dan tanggung jawab dengan bijaksana di antara anggota tim. Mereka harus memahami keahlian dan kekuatan individu dalam tim dan mendelegasikan tugas sesuai dengan kemampuan dan minat masing-masing anggota tim.

4. Pemecahan Masalah dan Pengambilan Keputusan

Seorang pemimpin proyek perangkat lunak dihadapkan pada berbagai tantangan dan keputusan yang perlu diambil. Mereka harus memiliki kemampuan dalam pemecahan masalah, evaluasi situasi dengan bijaksana, dan mengambil keputusan yang tepat dengan mempertimbangkan berbagai faktor dan dampaknya terhadap proyek.

5. Kemampuan Memotivasi Tim

Pemimpin proyek perangkat lunak yang efektif harus mampu memotivasi anggota tim untuk mencapai hasil terbaik. Ini melibatkan mengenali prestasi individu, memberikan penghargaan yang pantas, memberikan umpan balik yang konstruktif, dan menciptakan lingkungan kerja yang positif dan kolaboratif.

6. Kolaborasi dan Pekerjaan Tim

Pemimpin proyek perangkat lunak harus mendorong kolaborasi dan kerja tim yang efektif. Mereka harus memfasilitasi komunikasi dan kerjasama antara anggota tim, membangun kepercayaan, dan menciptakan budaya kerja yang inklusif dan saling mendukung.

7. Fleksibilitas dan Adaptabilitas

Dalam lingkungan pengembangan perangkat lunak yang cepat berubah, seorang pemimpin proyek perangkat lunak harus memiliki fleksibilitas dan kemampuan adaptasi. Mereka harus siap menghadapi perubahan kebutuhan proyek, mengatasi hambatan, dan mengarahkan tim menuju penyelesaian yang sukses.

Perilaku kepemimpinan yang efektif dalam proyek perangkat lunak memainkan peran penting dalam membangun tim yang kuat, mencapai tujuan proyek, dan memberikan hasil yang berkualitas.

Tim Proyek High Performance

Tim Proyek yang mencakup sejumlah elemen penting yang mencerminkan keberhasilan tim dalam mencapai tujuan proyek dengan kinerja yang tinggi. Berikut adalah penjelasan tentang setiap elemen tersebut:

1. Komunikasi Terbuka

Tim proyek menerapkan komunikasi terbuka yang memungkinkan anggota tim untuk berbagi informasi, gagasan, dan masalah dengan jujur dan transparan. Komunikasi terbuka membantu mengurangi hambatan komunikasi dan kerjasama, serta memperkuat hubungan tim.

2. Pemahaman Bersama

Tim proyek berusaha untuk mencapai pemahaman bersama tentang tujuan, strategi, dan harapan proyek. Pemahaman bersama memastikan bahwa semua anggota tim memiliki visi dan misi yang sama tentang apa yang ingin dicapai, dan meminimalkan kebingungan atau kesalahpahaman yang dapat mempengaruhi kinerja tim.

3. Kepemilikan Bersama

Kepemilikan bersama adalah prinsip yang mendasari tim proyek. Setiap anggota tim merasa memiliki tanggung jawab terhadap keberhasilan proyek dan bekerja sama untuk mencapai hasil yang diinginkan. Kepemilikan bersama membangun semangat tim dan menghasilkan komitmen yang kuat terhadap kinerja yang tinggi.

4. Trust (Kepercayaan)

Kepercayaan adalah faktor kritis dalam tim proyek. Anggota tim harus saling mempercayai, memegang komitmen mereka, dan menghormati kontribusi masing-masing. Kepercayaan memperkuat kerjasama, mengurangi konflik, dan mempercepat pengambilan keputusan.

5. Kolaborasi

Kolaborasi yang efektif adalah kunci dalam tim proyek. Anggota tim bekerja bersama, menggabungkan keahlian dan pengalaman mereka, dan saling melengkapi untuk mencapai tujuan proyek. Kolaborasi membangun dinamika tim yang kuat dan meningkatkan kemampuan tim untuk mengatasi tantangan yang kompleks.

6. Kemampuan Beradaptasi

Tim proyek memiliki kemampuan untuk beradaptasi dengan perubahan yang terjadi selama proyek. Mereka fleksibel dan terbuka terhadap perubahan

yang situasional, mampu mengatasi hambatan, dan menyesuaikan strategi mereka untuk mencapai hasil yang diinginkan.

7. Resilience (Ketahanan)

Ketahanan merupakan kemampuan tim untuk menghadapi tantangan, hambatan, dan kegagalan dengan penuh semangat dan kembali bangkit. Tim proyek memiliki ketahanan yang tinggi, dapat pulih dari kegagalan, dan menggunakan pengalaman tersebut untuk belajar dan tumbuh.

8. Empowerment (Pemberdayaan)

Tim proyek memberdayakan anggota tim dengan memberikan otoritas, kepercayaan, dan tanggung jawab yang tepat. Pemberdayaan memungkinkan anggota tim untuk mengambil inisiatif, membuat keputusan, dan berkontribusi secara signifikan terhadap kesuksesan proyek.

9. Recognition (Pengakuan)

Tim proyek memberikan pengakuan yang layak kepada anggota tim yang berkinerja baik dan berkontribusi secara positif. Pengakuan termasuk pujian, apresiasi, dan penghargaan yang memperkuat motivasi dan semangat kerja anggota tim.

Tim proyek yang menerapkan elemen-elemen ini memiliki peluang yang lebih baik untuk mencapai hasil yang diinginkan dengan efisiensi dan efektivitas yang tinggi.

BAB 10 PENGENDALIAN PROYEK

Pendahuluan

Proyek merupakan sekumpulan aktivitas yang menghasilkan barang atau jasa yang mempunyai tujuan tertentu atau unik yang cukup penting bagi pihak Manajemen atau Organisasi. Karena Proyek merupakan kegiatan yang penting bagi pihak Manajemen atau Organisasi sehingga diperlukan pengendalian. Pengendalian adalah suatu kegiatan pengawasan suatu proyek agar berjalan sesuai dengan rencana telah ditentukan dan hasil yang baik.

Selain mendapat hasil yang baik pengendalian juga menyangkut biaya dan waktu serta pengambilan langkah-langkah yang diperlukan selama pelaksanaan proyek. Sebagai salah satu fungsi dan proses kegiatan dalam manajemen proyek yang sangat berpengaruh terhadap hasil akhir proyek adalah pengendalian yang mempunyai tujuan utama untuk meminimalisir penyimpangan yang mungkin terjadi selama proses proyek berlangsung.

Juga mengecek apakah kegiatan atau aktivitas sudah sesuai dengan estimasi dan rencana awal, serta sudah sesuai dengan target atau belum. Bila belum sesuai target, maka perlu tindakan apa sajakah yang harus dilakukan agar tujuan proyek bisa terpenuhi.

Tujuan Pengendalian Proyek

Menurut PMBOK Manajemen Proyek dapat

diartikan sebagai “*is application of knowledge, skills, tools and techniques to project activities to meet project requirements*” atau aplikasi dari pengetahuan, keahlian, alat dan teknik untuk melaksanakan kegiatan sesuai dengan kebutuhan proyek. Kegiatan manajemen proyek meliputi perencanaan pekerjaan, menilai risiko, memperkirakan sumber daya yang diperlukan untuk menyelesaikan pekerjaan, pengorganisasian pekerjaan, memperoleh sumber daya manusia dan material, menetapkan tugas, kegiatan mengarahkan dan mengendalikan proyek, melaporkan kemajuan dan menganalisis hasil.

Adapun tujuan pengendalian proyek yaitu:

1. Memastikan pencapaian tujuan proyek sehingga sesuai dengan target yang telah ditentukan.
2. Mengontrol pelaksanaan proyek agar sesuai dengan estimasi dan rencana awal.
3. Adanya kontrol dilakukan, memberikan masukan apakah project management plan perlu di update ataukah tidak.

Mekanisme Pengendalian Proyek

Mekanisme Pengendalian terhadap semua pelaksanaan proyek, meliputi:

1. Pengontrolan terhadap waktu, batasan dan mutu.
2. Pengontrolan terhadap biaya.
3. Pembuatan Laporan tentang status dan kemajuan proyek.
4. Jika dari hasil pengontrolan diperlukan perubahan planning, maka lakukan re-planning.

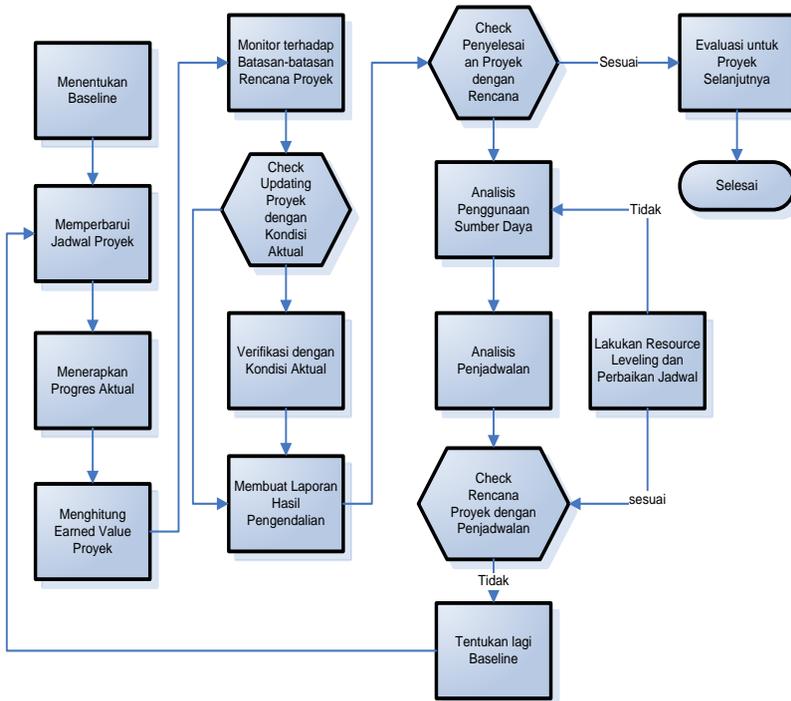
Pemantauan Kinerja Proyek

Tujuan pemantauan kinerja proyek adalah untuk mengetahui apakah proyek berjalan sebagaimana mestinya.

Manager proyek harus memiliki acuan dasar sebagai tujuan dan sasaran pengendalian (Husen,2011). Manajer Proyek (PM) melacak dimensi proyek (ruang lingkup, jadwal, biaya dan kualitas), memantau risiko, perubahan proyek dan kinerja proyek secara keseluruhan, dan harus dapat melaporkan dan meramalkan evolusi proyek kepada para pemangku kepentingan proyek. Informasi ini kemudian disediakan (didistribusikan) kepada para pemangku kepentingan yang relevan, sebagaimana didefinisikan dalam Rencana Manajemen Komunikasi.

Pengendalian Jadwal

Tujuan dari kontrol jadwal adalah untuk memastikan bahwa tugas-tugas proyek dilaksanakan sesuai jadwal dan bahwa tenggat waktu proyek tercapai. Manajer Proyek (PM) secara teratur memonitor jadwal dan melacak perbedaan antara aktivitas/tenggat waktu yang direncanakan, aktual dan perkiraan. Perubahan pada tugas (misalnya upaya yang diperlukan atau tanggal mulai/berakhir) yang berdampak pada keseluruhan proyek jadwal dikompilasi dan dimasukkan ke dalam Rencana Kerja Proyek (status jadwal yang diperbarui). Jika jadwal beresiko atau penundaan yang cukup besar diperkirakan, Komite Pengarah Proyek (Project Steering Committee/PSC) perlu diberitahu dan tindakan korektif perlu dirancang, disetujui dan dilaksanakan. Jika hal ini terjadi, para pemangku kepentingan proyek yang terkena dampak juga harus diberitahu. Diagram alir pengendalian Schduling Proyek terlihat pada Gambar 21 sebagai berikut:



Gambar 21. Diagram alir pengendalian Schduling Proyek (EDP Media, 2013)

Salah satu bagian dari tujuan utama pengendalian proyek adalah waktu. Keterlambatan waktu pelaksanaan proyek dapat mengakibatkan kerugian, seperti penambahan biaya, keterlambatan pemanfaatan produk, dan sebagainya. Batasan Biaya, mutu serta lingkup proyek merupakan aspek dalam pengendalian waktu, supaya proyek dapat diselesaikan sesuai atau lebih cepat dari rencana. Hal-hal yang perlu diperhatikan dalam pengendalian waktu proyek yaitu:

1. Identifikasi kegiatan
2. Penyusunan urutan kegiatan
3. Perkiraan kurun waktu
4. Penyusunan jadwal
5. Pengendalian waktu

6. Teknik dan metode

Teknik dan metode dalam pengendalian waktu atau jadwal, yaitu:

1. Bagan Balok dan Jaringan Kerja (CPM, PERT, PDM) untuk membuat jadwal dan menganalisis waktu penyelesaian proyek.
2. Database dan historical record untuk memperkirakan jangka waktu komponen aktivitas.
3. Resource leveling untuk meratakan penggunaan sumberdaya.
2. Cost and schedule trade off untuk menemukan jadwal yang ekonomis.
3. Simulasi, contohnya analisis Monte Carlo.
4. Fast tracking.

Pengendalian Biaya

Biaya proyek adalah salah satu indikator yang penting dan perlu dicermati pengendaliannya agar sesuai dengan anggaran yang telah dianggarkan sehingga dapat menghasilkan keuntungan proyek yang maksimal. Tujuan dari pengendalian biaya adalah untuk mengelola biaya proyek agar sesuai dengan biaya/upaya dasar dan batasan anggaran proyek secara keseluruhan. Manajer Proyek (PM) secara teratur memonitor anggaran dan melacak perbedaan antara biaya yang dianggarkan, aktual, dan perkiraan. Jika anggaran proyek beresiko, Komite Pengarah Proyek (*Project Steering Committee/PSC*) perlu diberitahu dan tindakan korektif perlu dirancang, disetujui dan dilaksanakan. Jika terjadi pembengkakan biaya yang cukup besar diperkirakan terjadi, hal ini harus dibenarkan, dilaporkan dan disetujui oleh Pemilik Proyek (*Project Owner (PO)*) atau Badan Tata Kelola yang Sesuai.

Gantt Chart

Gantt Chart ialah teknik manajemen proyek yang umum digunakan untuk merepresentasikan jadwal, fase, dan aktivitas proyek dalam satu visual (umumnya berupa diagram batang horizontal). Ini berfokus pada proyek urutan, durasi, ketergantungan, dan status proyek dengan cara yang mudah dipahami. Gantt Chart menunjukkan urutan kegiatan yang perlu dilakukan dan memberikan gambaran umum tentang kemajuan yang telah dicapai pada setiap titik waktu. Selain itu Gantt Chart memiliki kemampuan untuk menampilkan dengan jelas status setiap aktivitas secara sekilas.

Metode Jalur Kritis/*Critical Path Method* (CPM)

Metode Jalur Kritis (CPM) yaitu teknik pemodelan yang menggunakan algoritme berbasis matematika untuk menghitung durasi total sebuah proyek. CPM menghitung jalur terpanjang yang diperlukan (durasi terpanjang yang tidak dapat dihindari) dari kegiatan yang direncanakan dari awal hingga akhir proyek, atau dikenal sebagai jalur kritis proyek. Teknik ini membantu untuk memahami aktivitas mana yang memiliki pengaruh kritis terhadap keseluruhan durasi proyek. Karena jalur kritis mewakili jalur aktivitas terpanjang yang diperlukan, jalur ini juga mewakili durasi terpendek kemungkinan durasi proyek hingga selesai. Berdasarkan informasi ini, kegiatan dapat diprioritaskan untuk memperpendek durasi jalur kritis dengan memangkas aktivitas jalur kritis, melakukan lebih banyak kegiatan secara paralel atau menambahkan lebih banyak sumber daya.

Metode Rantai Kritis (CCM)

Metode Rantai Kritis (CCM) adalah teknik pemodelan yang digunakan untuk merencanakan dan menjadwalkan serangkaian kegiatan atau proyek. Metode ini mirip dengan Metode Jalur Kritis (CPM), tetapi memperhitungkan sumber daya dan levelnya, serta perilaku Manajer Proyek (PM) saat memperkirakan durasi aktivitas dalam sebuah proyek. Teknik ini didasarkan pada pengamatan bahwa estimasi waktu aktivitas untuk proyek hampir dua kali lipat dari waktu yang dibutuhkan untuk menyelesaikan kegiatan. Alasan yang menyebabkan keterlambatan dapat mencakup tidak memanfaatkan menyelesaikan aktivitas lebih awal, mondar-mandirnya anggota tim untuk mengisi waktu yang tersedia untuk menyelesaikan tugas, menunggu hingga saat terakhir untuk benar-benar fokus pada tugas yang sedang dikerjakan. CCM mengasumsikan bahwa perkiraan durasi kegiatan oleh Manajer Proyek sudah tepat, dan segera melanjutkan untuk mengurangnya.

Manajemen Nilai yang Diperoleh (EVM)

Earned Value Management (EVM) merupakan teknik yang digunakan untuk memantau dan mengendalikan kinerja proyek, memberikan pandangan obyektif tentang kinerja berdasarkan keuangan proyek. Baik biaya maupun nilai adalah diukur dalam bentuk unit biaya (misalnya hari orang atau euro). EVM menyediakan metrik yang relatif objektif – atau Indikator Kinerja Utama (KPI) - untuk mengelola kinerja proyek secara proaktif. Beberapa indikator merefleksikan kemajuan yang telah dicapai sejauh ini, atau penyimpangan dari rencana dari sudut pandang biaya atau nilai pekerjaan, sementara indikator lainnya Indikator ini berfokus pada perkiraan

total deviasi anggaran, atau pada produktivitas yang diperlukan untuk menyelesaikan proyek sesuai jadwal.

Pengelolaan Pemangku Kepentingan

Mengelola pemangku kepentingan proyek adalah aktivitas manajemen proyek yang penting yang dimulai sejak awal proyek siklus hidup proyek ketika ekspektasi dan persyaratan proyek diidentifikasi (dalam Fase Memulai), dan diakhiri dengan mencatat pengalaman dan kepuasan pemangku kepentingan secara keseluruhan (dalam Fase Penutupan). Tanggung jawab untuk kegiatan ini adalah Manajer Proyek (PM). Pengarah Proyek (Project Steering Committee/PSC) juga harus dilibatkan, khususnya Business Manager (BM) yang harus membantu mengelola pemangku kepentingan di sisi pemohon (misalnya pengguna).

Pengelolaan Persyaratan

Manajemen Persyaratan adalah proses mengumpulkan, mendokumentasikan, dan memvalidasi persyaratan, dan mengelola implementasi dan perubahannya. Ini adalah proses yang berjalan terus menerus sepanjang proyek siklus hidup proyek dan berhubungan dengan proses manajemen proyek lainnya, seperti manajemen kualitas dan perubahan. Proses Manajemen Persyaratan dapat disesuaikan dan disesuaikan dengan kebutuhan proyek dan dapat didokumentasikan baik dalam Rencana Manajemen Persyaratan atau dalam Buku Panduan Proyek. Terpisah Dokumen persyaratan yang terpisah digunakan untuk menentukan, mengkategorikan, dan memprioritaskan persyaratan. Dokumen-dokumen tersebut dapat berupa dokumen yang berdiri sendiri atau lampiran pada Piagam Proyek.

Pengelolaan Perubahan Proyek

Manajemen perubahan proyek mendefinisikan aktivitas yang terkait dengan mengidentifikasi, mendokumentasikan, menilai, memprioritaskan, menyetujui, merencanakan dan mengendalikan perubahan proyek, dan mengkomunikasikannya kepada semua pemangku kepentingan yang relevan. Perubahan dapat diminta (atau diidentifikasi dan diajukan) di sepanjang siklus hidup proyek oleh pemangku kepentingan proyek dan dapat dikaitkan dengan perubahan dalam ruang lingkup proyek, persyaratan, hasil dan fitur, karakteristik kualitas, atau pencapaian. Proses Manajemen Perubahan Proyek dapat disesuaikan dan disesuaikan dengan kebutuhan proyek dan dapat didokumentasikan dalam Rencana Manajemen Perubahan Proyek atau dalam Buku Panduan Proyek. Log Perubahan digunakan untuk memantau dan mengendalikan semua perubahan proyek. Hal ini membantu melacak perubahan dan mengomunikasikannya kepada Pemilik Proyek (PO) dan/atau Komite Pengarah Proyek (PSC) untuk mendapatkan persetujuan.

Pengelolaan Risiko

Manajemen risiko yaitu proses yang berkesinambungan dan sistematis untuk mengidentifikasi, menilai, dan mengelola risiko agar sejalan dengan selera risiko yang diterima. Manajemen risiko meningkatkan kemampuan dan kepercayaan diri tim proyek untuk menangani ketidakpastian. Manajemen risiko berfokus pada penanganan secara proaktif terhadap setiap kejadian yang dapat mengancam tujuan proyek. Proses Manajemen Risiko dapat disesuaikan dan disesuaikan dengan kebutuhan proyek dan dapat didokumentasikan baik dalam

Rencana Manajemen Risiko atau dalam Buku Panduan Proyek. Log Risiko digunakan untuk mendokumentasikan dan mengkomunikasikan risiko dan tindakan serta tanggung jawab respons risiko yang relevan.

Pengelolaan Masalah dan Keputusan

Manajer Proyek (PM) mengelola masalah dan keputusan proyek. Masalah diidentifikasi, dievaluasi dan ditugaskan untuk resolusi kepada pemangku kepentingan proyek yang relevan sesuai dengan proses Manajemen Isu, yang dapat didokumentasikan baik dalam Rencana Manajemen Isu atau dalam Buku Panduan Proyek. Log Masalah digunakan untuk mengelola masalah proyek, sedangkan Log Keputusan digunakan untuk mendokumentasikan semua keputusan yang relevan. Perhatikan bahwa masalah dan keputusan sering kali terkait dengan resolusi item log lainnya (mis. risiko, perubahan).

Pengelolaan Kualitas

Manajemen kualitas proyek bertujuan untuk memastikan bahwa proyek akan mencapai hasil yang diharapkan dengan cara yang paling efisien dan bahwa hasil kerja akan diterima oleh para pemangku kepentingan yang relevan. Hal ini melibatkan pengawasan semua kegiatan yang diperlukan untuk mempertahankan tingkat keunggulan yang diinginkan. Ini termasuk melakukan perencanaan kualitas, jaminan kualitas, kontrol kualitas dan peningkatan kualitas hingga penerimaan akhir proyek (dalam Fase Penutupan). Manajemen konfigurasi akan membantu para pemangku kepentingan proyek untuk mengelola artefak proyek dan hasil proyek secara efektif dan untuk memberikan referensi tunggal yang dapat

diandalkan kepada mereka, memastikan bahwa yang benar disampaikan kepada pemohon proyek/klien. Manajer Proyek (PM) harus memastikan bahwa tujuan, pendekatan, persyaratan, aktivitas, metrik, dan tanggung jawab proses manajemen mutu proyek didefinisikan dengan jelas dan didokumentasikan dalam Rencana Manajemen Mutu.

Pengelolaan Penerimaan Hasil Kerja

Sebuah proyek dapat menghasilkan satu atau beberapa hasil kerja. Setiap hasil kerja ini harus diterima secara resmi. Manajemen penerimaan hasil memastikan bahwa hasil ini memenuhi tujuan yang telah ditetapkan dan

Kriteria yang ditetapkan dalam Rencana Penerimaan Hasil sehingga pemohon proyek dapat menerimanya secara resmi. Perhatikan bahwa penerimaan proyek akhir terjadi di Fase Penutupan.

Pengelolaan Transisi

Manajemen transisi memastikan bahwa transisi dari kondisi lama ke kondisi baru (yaitu menggunakan baru) dilakukan dengan cara yang terkendali dan lancar. Hal ini mencakup pengelolaan segala sesuatu yang relevan aktivitas komunikasi yang relevan dan membutuhkan kerja sama yang erat dengan Business Manager (BM) untuk memastikan transfer hasil proyek yang benar ke organisasi klien.

Pengelolaan Implementasi Bisnis

Pelaksanaan yang efektif dari semua kegiatan implementasi bisnis sangat penting untuk kelancaran operasi bahkan setelah hasil proyek disampaikan kepada pemangku kepentingan/komunitas pengguna. Dengan cara

ini, bisnis kegiatan implementasi bisnis saling melengkapi dengan kegiatan transisi. Perhatikan bahwa kegiatan implementasi bisnis hampir selalu diperlukan setelah proyek berakhir, sehingga merupakan praktik yang baik untuk juga mendefinisikan kegiatan perubahan pasca proyek. Implementasi kegiatan ini berada di luar lingkup tanggung jawab proyek, dan oleh karena itu menjadi tanggung jawab organisasi permanen dan dilakukan sebagai bagian dari operasi yang sedang berlangsung atau proyek-proyek di masa depan.

Pengelolaan Alih Daya

Manajer Proyek (PM) bekerja sama dengan otoritas pengadaan yang relevan untuk menentukan yang paling tepat, menguraikan ruang lingkup produk dan/atau layanan yang akan dikontrakkan, dan bekerja dengan Manajer Proyek Kontraktor (CPM) untuk memastikan bahwa kontraktor memberikan kualitas pekerjaan yang diharapkan sebagaimana didefinisikan dalam Rencana Pengalihdayaan.

Daftar Pustaka

- A.S, R., & Shalahuddin, M. (2019). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Penerbit INFORMATIKA.
- Alla, B., Sergiy, B., Svitlana, O., & Tanaka, H. (2020). Entropy paradigm of project-oriented organizations management. *CEUR Workshop Proceedings*, 2565, 233–243.
- Amrit, C. (2005). Coordination in software development: The problem of task allocation. *Proceedings of the 2005 Workshop on Human and Social Factors of Software Engineering, HSSE 2005, April*. <https://doi.org/10.1145/1083106.1083107>
- Baker, F. T. (2010). Chief programmer team management of production programming. *IBM Systems Journal*, 11(1), 56–73. <https://doi.org/10.1147/sj.111.0056>
- Bartezzaghi, E., Corso, M. and Verganti, R. (1997) Continuous improvement and inter-project learning in new product development. *International Journal of Technology Management*.
- Boehm, B., & DeMarco, T. (1986). *Software Engineering Economics*. Prentice Hall.
- Cleland, D. I., & Ireland, L. R. (2006). *Project Management: Strategic Design and Implementation (5th Edition)*. McGraw-Hill Education.

dailysocial.id. (2022, desember 30). 5 contoh surat penawaran beserta formatnya, sangat bermanfaat. Diakses 1/7/2023. Dari <https://dailysocial.id/post/contoh-surat-penawaran>

DeMarco, T., & Lister, T. (1999). *Peopleware Productivity Project and Team* (2nd ed.). Dorset House.

Fleming, Q. W., & Koppelman, J. M. (2016). *Earned Value Project Management*. Project Management Institute.

ftware Projects. Addison-Wesley Professional.

Garmus, D., & Herron, D. (2001). *Function Point Analysis: Measurement Practices for Successful So*

Ghozali, K. (2004, Juli). *Rekayasa Resiko Pengembangan Perangkat Lunak*. *Jurnal Ilmiah Teknologi Informasi*, 2(2), 76-84.

Hazar H. (2014). *Time Management tools and techniques for project management*. *Socioeconomic Research Bulletin*, 62.

Heryanto I , Triwibowo T. 2013. “Manajemen Proyek Berbasis Teknologi Informasi”, *Informatika: Bandung*

Heryanto. I dan Triwibowo. T., 2013. “Manajemen Proyek berbasis Teknologi Informasi”.

<http://dx.doi.org/10.12962/j24068535.v3i2.a259>

<https://ejournal.atmajaya.ac.id/index.php/metris/article/view/2800/1494>

Husen. A., 2011. “Manajemen Proyek Perencanaan, Penjadwalan dan Pengendalian Proyek”.

- Ideutama, cv .2023. Proposal pembuatan elearning berbasis web bromo bravo college. Malang
- Institute, P. M. (2014). Project Management Institute Annual Report 2014. <http://www.pmi.org/default.aspx>
- IPMA. (2015). IPMA Individual Competence Baseline (ICB), Version 4.0. International Project Management Association.
- Jackman, M. (1998). Homeopathic Remedies for Team Toxicity. *IEEE Software*, 15(4), 43–45. <https://doi.org/10.1109/52.687943>
- Jahan, M. S., Talha Riaz, M., Arif, K. S., & Abbas, M. (2019). Software project management and its tools in practice in IT Industry of Pakistan. 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies, ICoMET 2019, May 2020, 1–6. <https://doi.org/10.1109/ICOMET.2019.8673535>
- Jørgensen, M., & Shepperd, M. (2007). A Systematic Review of Software Development Cost Estimation Studies. *IEEE Transactions on Software Engineering*, 33(1), 33-53.
- Kerzner, H. (2017). *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*. Wiley.
- Kundu, J., Bishoi, T. K., Bhattacharya, M., & Chowdhury, A. (2015). Project management – an overview. *International Journal of Current Innovation Research*,

1(4), 91–101. <https://doi.org/10.4324/9781003005216-1>

Lewis, J. P. (2018). *Project Planning, Scheduling & Control*. McGraw-Hill Education.

Mantei, M. (1981). The Effect of Programming Team Structures on Programming Tasks. *Communications of the ACM*, 24(3), 106–113. <https://doi.org/10.1145/358568.358571>

Marchewka, J. (2003). *Information technology project management- providing measurable organizational value*. John Wiley & Sons, Inc.

Mardiyanto, Fuad Fahmi & Fiqih Satria. (2021). Expert System For Diagnosing Diseases In Betta Fish Based On Android. *International Journal of Artificial Intelligence and Robotic Technology*,1(2), 35-44.

Martin, R. (1997). *Project Estimating and Cost Management*. Amacom.

McConnell, S. (2006). *Software Estimation: Demystifying the Black Art*. Microsoft Press.

Mendes, E., Mosley, N., & Counsell, S. (2012). A Systematic Review of Software Development Cost Estimation Studies. *IEEE Transactions on Software Engineering*, 38(2), 398-462.

Muller, R., & Turner, J. R. (2007). Matching the Project Manager ' s Leadership Profile to Project Type. *International Journal of Project Management* 25, 25,

21–32.

<https://doi.org/10.1016/j.ijproman.2006.04.003>

Nuris, A. M., Maharani, A., & Rachmadita, R. N. (2021). Analisis Risiko Proyek Pengembangan Perangkat Lunak Menggunakan Kerangka Kerja ISO 31000. *Jurnal Metris*, 22(2), 73-81.

Paliktzoglou, V. (2015). *Managing Projects with PRINCE2* (Issue February).

<https://doi.org/10.13140/2.1.3992.6567>

Pandian, C. R. (2006). *Applied Software Risk Management A Guide for Software Project Managers*. Boca Raton: Taylor & Francis Group.

Pengembangan Rekayasa Perangkat Lunak. *Teknika*, 5(1), <https://doi.org/10.34148/teknika.v5i1.48>

Pengertian, Sejarah, Fungsi, dan Contohnya. (7 juni 2023). <https://www.gramedia.com/literasi/pengetian-software>

PMBOK Guide.2004. A Guide to the project management body of knowledge 3rd edition. Project management Institute: Pennsylvania USA.

PMI. (2008). *A Guide to the Project Management Body of Knowledge*. Project Management Institute.

PMI. (2013). *A Guide to the Project Management Body of Knowledge* (5th ed.). Project Management Institute.

Pressman, R. S. (2010). *Software engineering: a practitioner's approach* (7th ed.). McGraw-Hill Book

- Company Inc. <https://doi.org/10.1145/336512.336521>
- Pressman, Roger. S. (2001). *Software Engineering: A Practitioner's Approach* (5th ed.). New York: McGrawHill.
- Project Management Institute (PMI). (2017). *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Sixth Edition*. PMI Publications.
- Project Management Institute. 2021. "A Guide to the Project Management Body of Knowledge (PMBOK Guide) and The Standard for Project Management". Seventh Edition. Project Management Institute, Inc.
- Rederp.co.id. (2023, mei 26). 3 contoh surat pengadaan barang dan manfaatnya. diakses tanggal 1/7/2023. Dari <https://rederp.co.id/blog/contoh-surat-pengadaan-barang/>
- Ribeiro, J. L., & Martins, F. M. (2014). A review of software cost estimation in agile projects. *Information and Software Technology*, 56(6), 623-634.
- Rosmawarni, E. H., Ilhamsyah, & Mutiah, N. (2019). Audit Manajemen Risiko Pengembangan Perangkat Lunak Dengan Menggunakan Framework Cobit 5. *Coding: Jurnal Komputer dan Aplikasi*, 7(1), 54-63.
- Sawyer, S. (2004). Software development teams. *Communications of the ACM*, 47(12), 95–99. <https://doi.org/10.1145/1035134.1035140>
- Schwaber, Ken. 2004. "Agile Project Management with

- Scrum". Microsoft Press.
- Schwalbe, K. (2016). *Information Technology Project Management* (0 ed.). Cengage Learning.
- Smith, N. J., & Reinertsen, D. G. (2017). *Developing Products in Half the Time: New Rules, New Tools* (3rd Edition). Wiley.
- Sommerville, I. (2003). *Software Engineering* (Rekayasa Perangkat Lunak)/Edisi 6/Jilid 1 (Y. Hanum, Trans.). (H. W. Hardani, Ed.) Jakarta: Erlangga.
- Sudarma, Made. 2012. "Manajemen Proyek Teknologi Informasi". Udayana University Press.
- Sudhakar, G. P. (2012). A model of critical success factors for software projects. *Journal of Enterprise Information Management*, 25(6), 537–558. <https://doi.org/10.1108/17410391211272829>
- Tantra, Rudy. 2012. "Manajemen Proyek Sistem Informasi". Andi offset: Yogyakarta.
- The PM² Project Management Methodology Guide – Open Edition. V, 0, 9 November 2016
- Weinberg, G. M. (2016). *Becoming a Technical Leader : An Organic Problem-Solving Approach* (First). Dorset House Publishing.
- Widianto, Mardiyanto & Sobari. (2020). Implementation of Fingerprint Absence System and Web-Based Reporting at Roworejo Inposites Center, Negeri Katon Sub District. *JTKSI (Jurnal Teknologi Komputer dan Sistem*

Informasi), 3(3), 105-110.

Widjaya, Iwan Kurniawan.2013.”Manajemen Proyek teknologi Informasi”. Graha Ilmu: Yogyakarta.

Tentang Penulis



Suyono, S.Kom., M.T.I. Merupakan putra lahiran Pringsewu, 07 September 1990. Beliau menempuh pendidikan SD N 5 Pringsewu Barat, MTs N 1 Pringsewu, SMK KH. GHALIB Pringsewu, S-1 STMIK Pringsewu, S-2 IBI DARMAJAYA Bandar Lampung. Selain memiliki peran aktif di dalam dunia pendidikan beliau juga aktif menulis di beberapa jurnal ilmiah Nasional bereputasi dibidang Teknologi Komputer. Saat ini penulis juga aktif menjadi pembicara dan narasumber di berbagai pertemuan ilmiah Nasional dibidang Teknologi Komputer. Selain menghasilkan karya ilmiah beliau juga aktif di berbagai organisasi profesi Dosen anggota APTIKOM Lampung, anggota Perkumpulan Ahli dan Dosen Republik Indonesia (ADRI) Lampung.



Mardiyanto., S.Kom., M.T.I Lahir di Margo mulyo, 06 Oktober 1991. Penulis memulai pendidikan di SDN 03 Terbanggi Besar Lampung Tengah tahun 1998. Merupakan alumni SMPN 5 Terbanggi Besar Lampung Tengah Lulus tahun 2004, dan SMKN 2 Terbanggi Besar Lampung Tengah lulus tahun 2010. Setelah tamat SMK, penulis melanjutkan ke D3 mengambil jurusan Manajemen Informatika di STMIK Pringsewu Lampung lulus tahun 2013,”. Pada tahun 2015 penulis melanjutkan studi s1 di STMIK Pringsewu Lampung lulus tahun 2016.

Pada tahun 2016 bulan desember penulis melanjutkan ke jenjang S2 dengan mengambil jurusan Teknik Informatika di Institut Informatika dan Bisnis Darmajaya (IBI DARMAJAYA). Sejak kuliah penulis memiliki ketertarikan dengan bidang kajian yang berhubungan dengan dunia Pendidikan. Buku ini adalah salah satu karya dan semoga kedepannya secara konsisten akan disusul dengan buku-buku berikutnya. Pokok bahasan buku yang ditulis semata-mata untuk berbagi ilmu pengetahuan.



Dr. Jarudin, M.M., M.Pd., buku ini adalah salah satu karya dan inshaa allah secara konsisten akan disusul dengan buku-buku berikutnya. Pokok bahasan buku yang ditulis semata-mata untuk berbagi ilmu pengetahuan.



Tri Susilowati, NIDN: 0210037902, Perguruan tinggi: Institut Bakti Nusantara (IBN), Fakultas: Komputer, Pendidikan: S2 Teknik Informatika, Alamat: Pajaresuk, Pringsewu lampung.



Akrim Teguh Suseno, menyelesaikan program sarjana Teknik Informatika tahun 2013 di Universitas Dian Nuswantoro Semarang kemudian melanjutkan pendidikan dengan meraih program magister Teknologi Informasi tahun 2020 di Universitas Indonesia. Pada profesi karir pernah menjadi praktisi dibidang ICT selama lebih dari 5 tahun kemudian menjadi dosen tetap pada program studi Teknologi Komputer Institut Teknologi dan Sains Nahdlatul Ulama (ITSNU) Pekalongan. Selain itu penulis juga menjadi Kepala Prodi Teknologi Komputer di ITSNU Pekalongan dan fokus bidang peneliti saat ini adalah Data Science.



Linda Suvi Rahmawati, lahir di kota kediri, 17 Maret 1986. Penulis memulai Pendidikan di SDN Susuhbango Kecamatan Ringinrejo kabupaten kediri pada tahun 1992. Kemudian pada tahun 1998 melanjutkan ke SLTPN 1 Kandat kecamatan kandat kabupaten kediri. Dan tahun 2004 lulus dari SMUN₃ Kediri Kota kediri. Kemudian penulis melanjutkan ke jenjang perguruan tinggi yaitu kuliah di STMIK PPKIA Pradnya Paramita dengan jurusan D3 Manajemen Informatika dan melanjutkan ke jenjang S1 teknik Informatika di perguruan tinggi yang sama pada tahun 2007. Kemudian pada tahun 2013 penulis melanjutkan ke jenjang S2 di Universitas Gunadarma dengan jurusan Perangkat Lunak Sistem Informasi (PLSI). Sampai saat ini penulis aktif sebagai dosen di STMIK PPKIA Pradnya Paramita. Buku ini adalah salah satu karya dan inshaa allah secara konsisten akan disusul dengan buku-buku berikutnya. Pokok bahasan buku yang ditulis semata-mata untuk berbagi ilmu pengetahuan.



Frhendy Aghata, asli Surabaya yang menetap sekarang di kota Mojokerto bersama istri, pendidikan terakhir S1 (sarjana komputer) STMIK ASIA Malang/ITB ASIA Malang (sekarang) sebelum menjadi dosen sempat bekerja di beberapa perusahaan antara lain PT Sarana Kreasi Lestari, PT Solusi Bisnis Perkakas di daerah kawasan industri Surabaya dan Pasuruan, setelah itu melanjutkan S2 (teknologi informasi) di ISTTS Surabaya sambil mengajar di kampus IKHAC Mojokerto sampai saat ini serta membuka CV Ideutama bersama kawan bergerak di bidang pengembangan teknologi informasi bisa kunjungi website ideutama.com



Ibnu Atho'illah, S.T., M.T. Lahir di Pasuruan, 20 Agustus 1975. Penulis memulai Pendidikan di SD Nahdlatul Ulama Pasuruan Jawa Timur tahun 1982. Alumni SMPN 2 Pasuruan tahun 1991, dan SMAN 1 Pasuruan tahun 1994. Setelah tamat SMA, penulis melanjutkan Pendidikan S1 di Institut Teknologi Nasional (ITN) Malang Jurusan Teknik Kimia lulus tahun 1999. Pada tahun yang sama melanjutkan studi S2 di Universitas Gadjah Mada (UGM) di jurusan yang sama. Setelah lulus pada tahun 2002 sempat bekerja di PT. Kutrindo Indonesia (2002 – 2006). Selepas tahun 2006 penulis banyak berkecimpung di dunia Pendidikan, Pengajaran dan Pelatihan. Pada tahun 2008 – sekarang, penulis merupakan dosen pengajar di STMIK Bandung Bali, juga di tahun 2011 – sekarang sebagai Instruktur IT di LPK Emerald Informatika Bali.

buku ini adalah salah satu karya dan Insya Allah secara konsisten akan disusul dengan buku-buku berikutnya. Pokok bahasan buku yang ditulis semata-mata untuk berbagi ilmu pengetahuan.



Dr. Weda Adistianaya Dewa, S.Kom., MMSI, buku ini adalah salah satu karya dan nantinya secara konsisten akan disusul dengan buku-buku berikutnya. Pokok bahasan buku yang ditulis semata-mata untuk berbagi ilmu pengetahuan.



Dalam era di mana perangkat lunak telah merasuk ke hampir setiap aspek kehidupan kita, keberhasilan proyek pengembangan perangkat lunak menjadi semakin penting. "Manajemen Proyek Perangkat Lunak" adalah panduan komprehensif yang membawa pembaca melalui perjalanan mendalam dalam dunia mengelola proyek pengembangan perangkat lunak dengan sukses. Buku ini membuka tirai dengan memperkenalkan konsep dasar manajemen proyek dan relevansinya dalam konteks pengembangan perangkat lunak. Dari pengembangan perangkat lunak skala kecil hingga proyek yang kompleks, pembaca akan belajar bagaimana merencanakan, mengorganisir, dan mengawasi setiap tahap dengan cermat.



Dalam dunia yang semakin terhubung dan berubah dengan cepat, buku ini mengakui pentingnya melibatkan klien dan pemangku kepentingan dalam seluruh proses. Pembaca akan mendapatkan wawasan tentang bagaimana memahami kebutuhan mereka, melibatkan mereka dalam pengambilan keputusan, dan memastikan kepuasan akhir. Dengan pandangan luas dan pendekatan praktis, "Manajemen Proyek Perangkat Lunak" adalah panduan penting bagi para profesional IT, pengembang perangkat lunak, dan manajer proyek. Buku ini memberikan kombinasi yang baik antara teori dan aplikasi dunia nyata, membantu pembaca memahami bagaimana mengatasi tantangan yang terkait dengan proyek pengembangan perangkat lunak dan mencapai hasil yang sukses.

DITERBITKAN OLEH
PT. MIFANDI MANDIRI DIGITAL



Jln Payanibung Ujung D
Dalu Sepuluh-B, Tanjung Morawa
Kab. Deli Serdang Sumatera Utara

