

# REKAYASA RESIKO PENGEMBANGAN PERANGKAT LUNAK

**Khakim Ghozali**

Program Studi Sistem Informasi  
Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember  
Kampus ITS, Jl. Raya ITS, Sukolilo – Surabaya 60111, Telp. + 62 31 5939214, Fax. + 62 31 5913804  
Email : khakim@its-sby.edu

## ABSTRAK

*Pada makalah ini akan dibahas mengenai bagaimana mengelola resiko yang akan terjadi pada pengembangan perangkat lunak jika pengembangan tidak dapat berjalan sesuai dengan rencana. Pembahasan akan meliputi beberapa hal mulai dari bagaimana kita bisa mengidentifikasi resiko-resiko yang akan terjadi pada sebuah pengembangan perangkat lunak. Setelah kita mengetahui berbagai jenis resiko yang akan dihadapi maka berikutnya adalah membuat kategori dan prioritas tindakan untuk mengurangi atau menghilangkan resiko. Berikutnya adalah dilakukan perhitungan pengaruh resiko terhadap jadwal pengembangan.*

**Kata kunci :** rekayasa resiko

## 1. PENDAHULUAN

Pada saat kita mengerjakan pengembangan perangkat lunak sering kita menghadapi berbagai situasi yang tidak nyaman seperti keterlambatan pengembangan atau pengeluaran biaya pengembangan yang melebihi anggaran. Hal ini dikarenakan kurang siapnya kita menghadapi berbagai kemungkinan resiko yang akan terjadi. Untuk itu perlu dilakukan identifikasi tindakan yang harus dilakukan untuk mencegah ataupun meminimalkan resiko tersebut.

Beberapa resiko lebih penting dibandingkan resiko lainnya. Baik penting maupun tidak sebuah resiko tertentu bergantung pada sifat resiko tersebut, pengaruhnya pada aktifitas tertentu dan kekritisan aktifitas tersebut. Aktifitas beresiko tinggi pada jalur kritis pengembangan biasanya merupakan penyebabnya.

Untuk mengurangi bahaya tersebut maka harus ada jaminan untuk meminimalkan resiko atau paling tidak mendistribusikannya selama pengembangan tersebut dan idealnya resiko tersebut dihapus dari aktifitas yang mempunyai jalur yang kritis.

## 2. TIPE RESIKO

Untuk keperluan identifikasi dan mengelola resiko yang bias menyebabkan sebuah pengembangan melampaui batas waktu dan biaya yang sudah dialokasikan maka perlu diidentifikasi tiga tipe resiko yang ada yaitu :

- Resiko yang disebabkan karena kesulitan melakukan estimasi
- Resiko yang disebabkan karena asumsi yang dibuat selama proses perencanaan.

- Resiko yang disebabkan adanya even yang tidak terlihat (atau tidak direncanakan).

### 2.1. Kesalahan Estimasi

Beberapa pekerjaan lebih sulit untuk melakukan estimasi dibandingkan pekerjaan lainnya disebabkan karena terbatasnya pengalaman pada pekerjaan serupa atau disebabkan karena jenis pekerjaan tersebut. Pembuatan sebuah user manual merupakan langkah yang tepat yang dapat dipertanggungjawabkan dan sebagai bukti bahwa kita pernah mengerjakan tugas yang serupa sebelumnya. Dengan pengalaman itu seharusnya kita mampu untuk melakukan estimasi dengan lebih tepat mengenai berapa lama pekerjaan dapat diselesaikan dan berapa besarnya biaya yang dibutuhkan.

Estimasi dapat ditingkatkan melalui analisa data histories untuk aktifitas yang serupa dan untuk system yang serupa. Dengan menyimpan perbandingan antara estimasi semula dengan hasil akhir akan mengakibatkan beberapa tipe pekerjaan sulit diestimasi secara tepat.

### 2.2. Asumsi Perencanaan

Pada setiap tahapan perencanaan, asumsi perlu dibuat, jika tidak benar maka dapat mengakibatkan resiko tersebut beresiko. Misal pada jaringan aktifitas, aktifitas dibangun berdasarkan pada asumsi menggunakan metodologi desain tertentu dimana memungkinkan urutan aktifitas diubah . Kita biasanya membuat asumsi bahwa setelah coding , biasanya sebuah modul akan diuji dan kemudian diintegrasikan dengan modul lainnya. Akan tetapi kita tidak merencanakan pengujian modul yang dapat

mengakibatkan perubahan desain awal. Hal ini dapat terjadi setiap saat.

Pada setiap tahapan pada proses perencanaan, sangat penting untuk memperinci secara eksplisit semua asumsi yang dibuat dan mengidentifikasi apa pengaruhnya jika ternyata dalam pelaksanaannya tidak sesuai dengan yang sudah direncanakan.

**2.3. Kemungkinan**

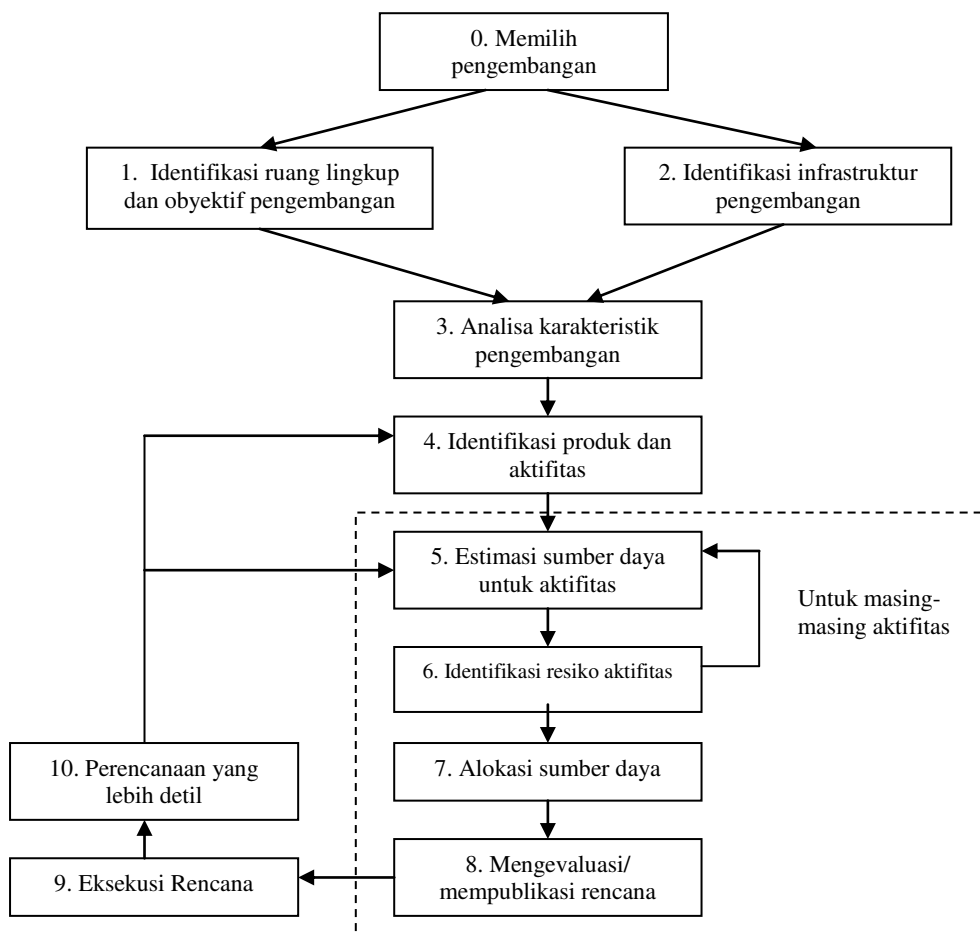
Beberapa kemungkinan dapat saja tidak pernah terlihat dan kita hanya dapat meyakinkan diri kita sendiri bahwa ada sesuatu yang tidak dapat dibayangkan, kadang-kadang dapat terjadi. Akan tetapi biasanya jarang terjadi hal seperti itu. Mayoritas kejadian yang tidak diharapkan biasanya dapat diidentifikasi – beberapa spesifikasi kebutuhan kemungkinan diubah setelah beberapa modul telah dikodekan, programmer senior meninggalkan pengembangan, perangkat keras yang diperlukan tidak dikirim tepat waktu. Beberapa kejadian

semacam itu dapat terjadi sewaktu-waktu dan walaupun kejadian tersebut kemungkinan terjadinya relatif rendah akan tetapi kejadian tersebut perlu dipertimbangkan dan direncanakan.

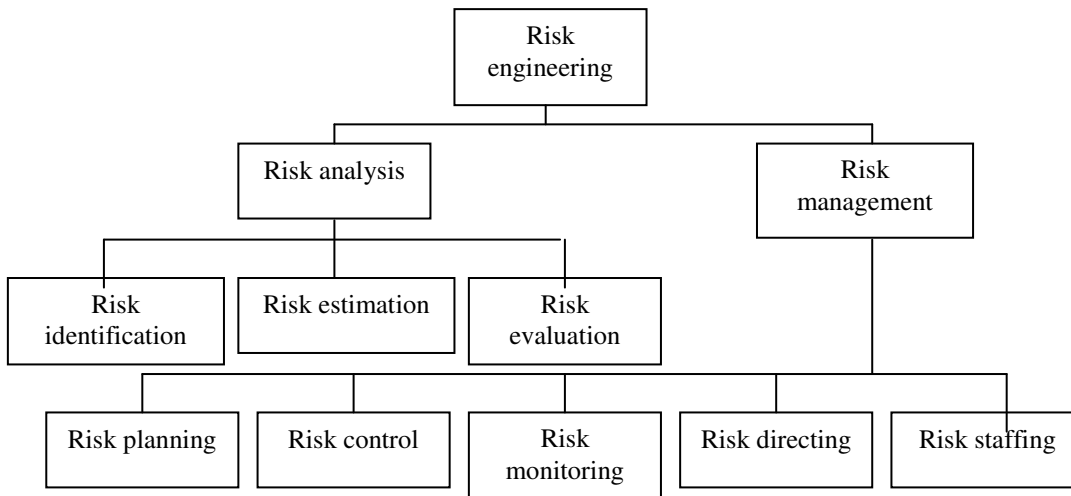
**3. MENGELOLA RESIKO**

Obyektif manajemen resiko adalah mencegah atau meminimisasi pengaruh yang tidak baik akibat kejadian yang tidak terduga melalui menghindari resiko atau mempersiapkan rencana kontingensi yang berkaitan dengan resiko tersebut.

Ada sejumlah model manajemen resiko, akan tetapi kebanyakan serupa yaitu dibedakan menjadi dua komponen utama – *identifikasi resiko dan manajemen resiko*. Contoh model yang sering dipergunakan terlihat pada gambar 2 yang memperlihatkan sebuah struktur yang memerinci aktifitas yang disebut oleh Barry Boehm dengan rekayasa resiko (*risk engineering*).



**Gambar 1. Analisis resiko dilaksanakan pada langkah 3 dan 6**



**Gambar 2. Struktur aktifitas rekayasa resiko versi Boehm**

***Risk identification***

Menjelaskan daftar semua resiko yang dapat mempengaruhi keberhasilan pelaksanaan pengembangan.

***Risk estimation***

Menjelaskan semua kemungkinan dan pengaruh yang akan terjadi pada masing-masing resiko.

***Risk evaluation***

Menjelaskan tingkatan resiko dan menentukan strategi menghindari resiko.

***Risk planning***

Menjelaskan rencana kontingensi dan dimana rencana tersebut yang paling tepat dipergunakan, menambahkan rencana tersebut pada struktur aktifitas pengembangan tersebut. Dalam pengembangan yang kecil, perencanaan resiko sepertinya merupakan tanggung jawab dari manajer pengembangan akan tetapi pada pengembangan yang medium atau pengembangan yang besar akan lebih bermanfaat ditunjuk seorang manajer khusus yang menangani resiko tersebut secara full-time.

***Risk control***

Berhubungan dengan fungsi utama manajer resiko untuk meminimisasi dan mengambil tindakan jika terjadi persoalan selama pengembangan tersebut berlangsung. Fungsi ini melibatkan aspek pengendalian kualitas selain yang berkaitan dengan berbagai problem pada saat resiko tersebut terjadi.

***Risk monitoring***

Harus menjadi sebuah aktifitas yang dijalankan karena jika terjadi resiko tertentu dapat mengubah pengembangan yang sedang berlangsung.

***Risk directing dan risk staffing***

Berkaitan dengan manajemen resiko sehari-hari. Strategi mencegah resiko dan menyelesaikan problem sering melibatkan tambahan staf , untuk itu harus direncanakan dan diarahkan.

**4. IDENTIFIKASI RESIKO**

Tahapan pertama dalam melakukan analisis resiko adalah mengidentifikasi bahaya yang dapat mempengaruhi durasi atau sumber daya pembiayaan pengembangan tersebut. Yang dikatakan bahaya disini adalah suatu keadaan yang dapat dan akan terjadi , dan jika keadaan muncul, dapat menciptakan suatu problem terhadap keberhasilan penyelesaian pengembangan.

Sebagai contoh, sakitnya salah seorang anggota tim merupakan bahaya yang dapat menimbulkan problem berupa terlambatnya penyerahan sebuah komponen. Terlambatnya penyerahan komponen tersebut dapat berpengaruh terhadap aktifitas lain. Dan jika aktifitas ini berada pada jalur kritis bisa berakibat pada terlambatnya penyelesaian pengembangan tersebut.

Sebuah cara umum untuk mengidentifikasi bahaya adalah dengan mempergunakan daftar checklist yang berisi semua kemungkinan bahaya yang bisa terjadi dan faktor yang mempengaruhi. Checklist semacam itu dapat berisi banyak faktor bahkan bisa mencapai ratusan jumlahnya.

Beberapa bahaya tersebut merupakan *generic risk* yang berarti bahwa bahaya tersebut relevan untuk semua pengembangan perangkat lunak dan standard checklist dapat dipergunakan berdasar hasil analisis pengembangan sebelumnya. Checklist ini berisi berbagai resiko seperti kesalahpahaman

terhadap kebutuhan atau personil kunci sedang sakit. Checklist tersebut juga berisi *specific risk* yang relevan untuk pengembangan tertentu dan hal ini akan terlihat lebih sulit untuk melakukan identifikasi tanpa melibatkan anggota dari tim pengembangan.

Beberapa kategori faktor yang perlu dipertimbangkan adalah sebagai berikut:

#### ***Application factor***

Sesuatu yang alami dari aplikasi – baik aplikasi pengolahan data yang sederhana, sebuah sistem kritis yang aman maupun sistem terdistribusi yang besar dengan elemen real-time terlihat menjadi sebuah faktor kritis. Ukuran yang diharapkan dari aplikasi juga sesuatu yang penting – sistem yang lebih besar, lebih besar dari problem error, komunikasi dan manajemennya.

#### ***Staff factors***

Pengalaman dan kemampuan staf yang terlibat merupakan faktor utama – seorang programmer yang berpengalaman, diharapkan akan sedikit melakukan kesalahan dibandingkan dengan programmer yang sedikit pengalamannya. Akan tetapi kita harus juga mempertimbangkan ketepatan pengalaman tersebut-pengalaman membuat modul dengan Cobol bisa mempunyai nilai kecil jika kita akan mengembangkan sistem kendali real-time yang kompleks dengan menggunakan C++.

Beberapa faktor seperti tingkat kepuasan staf dan tingkat pergantian dari staf juga penting untuk keberhasilan sebarang pengembangan – staf yang tidak termotivasi atau personil utama keluar dapat menyebabkan kegagalan pengembangan.

#### ***Project factors***

Merupakan hal yang penting bahwa pengembangan dan obyektifnya terdefinisi dengan baik dan diketahui secara jelas oleh semua anggota tim dan semua stakeholder utama. Jika hal ini tidak terlaksana dapat muncul resiko yang berkaitan dengan keberhasilan pengembangan tersebut. Dengan cara serupa, perencanaan kualitas yang formal dan telah disepakati harus dipahami oleh semua partisipan. Jika perencanaan kualitas kurang baik dan tidak tersosialisasi maka dapat mengakibatkan gangguan pada pengembangan tersebut.

#### ***Project methods***

Dengan mempergunakan spesifikasi dan metode terstruktur yang baik pada manajemen pengembangan dan pengembangan sistem akan mengurangi resiko penyerahan sistem yang tidak memuaskan atau terlambat. Akan tetapi penggunaan metode tersebut untuk pertama kali dapat mengakibatkan problem dan delay.

#### ***Hardware/software factors***

Sebuah pengembangan yang memerlukan hardware baru untuk pengembangan mempunyai resiko yang lebih tinggi dibandingkan dengan software yang dapat dibangun pada hardware yang sudah ada (dan familiar). Sebuah sistem yang dikembangkan untuk satu jenis hardware atau software platform tertentu jika dipergunakan pada hardware atau software platform lainnya bisa menimbulkan resiko tambahan (dan tinggi) pada saat instalasi.

#### ***Changeover factors***

Kebutuhan perubahan “all-in-one” kedalam suatu sistem baru mempunyai resiko tertentu. Perubahan secara bertahap atau gradual akan meminimisasi resiko akan tetapi cara tersebut tidak praktis. Menjalankan secara paralel dapat memberikan solusi yang aman akan tetapi biasanya tidak mungkin atau terlalu mahal.

#### ***Supplier factors***

Suatu pengembangan yang melibatkan organisasi eksternal yang tidak dapat dikendalikan secara langsung dapat mempengaruhi keberhasilan pengembangan. Misal tertundanya instalasi jalur telpon atau pengiriman peralatan yang sulit dihindari-dapat berpengaruh terhadap keberhasilan pengembangan.

#### ***Environment factors***

Perubahan pada lingkungan dapat mempengaruhi keberhasilan pengembangan. Misal terjadi perubahan regulasi pajak, akan mempunyai dampak yang cukup serius pada pengembangan aplikasi penggajian.

#### ***Health and safety factors***

Ada satu isu utama yaitu faktor kesehatan dan keamanan dari partisipan yang terlibat dalam pengembangan software walaupun tidak umum (dibandingkan dengan pengembangan teknik sipil) yang dapat mempengaruhi aktifitas pengembangan.

Meskipun beberapa faktor dapat mempengaruhi pengembangan secara keseluruhan maka perlu mempertimbangkan faktor tersebut secara individu pada masing-masing aktifitas – Misal sakitnya staf anggota kunci selama masa survey tidak terlalu berakibat serius dibandingkan ketidakhadirannya selama pelatihan user .

## **5. ANALISIS RESIKO**

Setelah resiko yang dapat mempengaruhi pengembangan teridentifikasi maka diperlukan cara untuk menentukan tingkat kepentingan dari masing-masing resiko. Beberapa resiko secara relatif tidak terlalu fatal (misal resiko keterlambatan penyerahan dokumentasi) sedangkan beberapa resiko lainnya berdampak besar. (misal resiko keterlambatan

penyerahan software). Beberapa resiko sering terjadi (salah satu anggota tim sakit sehingga tidak bisa bekerja selama beberapa hari). Sementara itu resiko lainnya jarang terjadi (misal kerusakan perangkat keras yang dapat mengakibatkan sebagian program hilang).

Probabilitas terjadinya resiko sering disebut dengan *risk likelihood*; sedangkan dampak yang akan terjadi jika resiko tersebut terjadi dikenal dengan *risk impact* dan tingkat kepentingan resiko disebut dengan *risk value* atau *risk exposure*. Risk value dapat dihitung dengan formula :

$$Risk\ exposure = risk\ likelihood \times risk\ impact$$

Idealnya risk impact diestimasi dalam batas moneter dan likelihood dievaluasi sebagai sebuah probabilitas. Dalam hal ini risk exposure akan menyatakan besarnya biaya yang diperlukan berdasarkan perhitungan analisis biaya manfaat. Risk exposure untuk berbagai resiko dapat dibandingkan antara satu dengan lainnya untuk mengetahui tingkat kepentingan masing-masing resiko.

Akan tetapi, estimasi biaya dan probabilitas tersebut sulit dihitung, subyektif, menghabiskan waktu dan biaya. Untuk mengatasi hal ini maka diperlukan beberapa pengukuran yang kuantitatif untuk menilai risk likelihood dan risk impact, karena tanpa ini sulit untuk membandingkan atau meranking resiko tersebut untuk berbagai keperluan.

Beberapa manajer resiko mempergunakan sebuah metode penilaian yang sederhana untuk menghasilkan ukuran yang kuantitatif pada saat mengevaluasi masing-masing resiko. Beberapa manajer memberikan kategori pada likelihood dan impact dengan high, medium atau low. Akan tetapi bentuk ini tidak memungkinkan untuk menghitung risk exposure. Sebuah pendekatan yang lebih baik dan populer adalah memberikan skor pada likelihood dan impact dengan skala tertentu misal 1-10. Jika suatu resiko kemungkinan besar akan terjadi diberi skor 10, sedangkan jika kecil jika kemungkinan terjadinya kecil maka akan diberi nilai 1.

Penilaian likelihood dan impact dengan skala 1-10 relatif mudah, akan tetapi kebanyakan manajer resiko akan berusaha untuk memberikan skor yang lebih bermakna, misal skor likelihood 8 akan dipertimbangkan dua kali likelihood dengan skor 4.

Hasil pengukuran impact, dapat diukur dengan skor yang serupa, harus dimasukkan pada perhitungan total risk dari proyek tersebut. Untuk itu harus melibatkan beberapa biaya potensial seperti :

- Biaya yang diakibatkan keterlambatan penyerahan atas jadwal yang sudah ditentukan
- Biaya yang berlebihan dikarenakan harus menambah sumber daya atau dikarenakan mempergunakan sumber daya yang lebih mahal.
- Biaya yang tidak terlihat pada beberapa komponen kualitas atau fungsionalitas sistem.

Tabel 1 berikut ini memperlihatkan contoh hasil evaluasi nilai resiko. Perhatikan bahwa resiko yang bernilai tertinggi tidak selalu akan menjadi resiko yang pasti terjadi maupun akan menjadi resiko dengan potensi impact yang terbesar.

**Tabel 1. Contoh evaluasi nilai risk exposure**

	Hazard	L	I	R
R1	Perubahan spesifikasi kebutuhan selama coding	1	8	8
R2	Spesifikasi perlu lebih lama dibandingkan yang diperlukan	3	7	21
R3	Staf sakit yang berpengaruh pada aktifitas yang kritis	5	7	35
R4	Staf sakit yang berpengaruh pada aktifitas yang tidak kritis.	10	3	30
R5	Pengkodean modul lebih lama dibandingkan yang diharapkan	4	5	20
R6	Pengujian modul memperlihatkan kesalahan atau ketidakefisiensian dalam desain.	1	10	10

### Prioritas resiko

Pengelolaan resiko melibatkan penggunaan dua strategi :

- Risk exposure dapat dikurangi dengan mengurangi likelihood atau impact
- Pembuatan rencana kontingensi berkaitan dengan kemungkinan resiko yang akan terjadi.

Sebarang usaha untuk mengurangi sebuah risk exposure atau untuk melaksanakan sebuah rencana kontingensi akan berhubungan dengan biaya yang berkaitan dengan usaha tersebut. Merupakan hal yang penting untuk menjamin bahwa usaha ini dilaksanakan dengan cara yang paling efektif dan diperlukan cara untuk memprioritaskan resiko sehingga usaha yang lebih penting dapat menerima perhatian yang lebih besar.

Estimasi nilai likelihood dan impact dari masing-masing usaha tersebut akan menentukan nilai risk exposure. Setelah risk exposure dapat dihitung maka resiko dapat diberi prioritas high, medium atau low sesuai dengan besar kecilnya nilai risk exposure.

Risk exposure yang berdasarkan pada metode penilaian perlu diberikan beberapa perhatian. Hasil evaluasi pada tabel 1, contoh, tidak memperlihatkan resiko R5 adalah dua kali lebih penting dibandingkan R6. Pada kasus ini, kita tidak bisa menginterpretasikan nilai risk exposure secara kuantitatif disebabkan nilai tersebut didasarkan pada metode penilaian yang non-cardinal. Pada kasus kedua, nilai exposure yang terlalu berjauhan akan mampu untuk membedakan antara resiko tersebut.

Akan tetapi risk exposure akan memungkinkan kita untuk memperoleh suatu ranking sesuai dengan kepentingannya. Pertimbangkan resiko pada tabel 1,

R3 dan R4 merupakan resiko yang paling penting dan kita dapat mengklasifikasikannya dengan high risk. Tingkat kepentingan yang berbeda dapat membedakan antara skor exposure satu dan dua ini dengan exposure tertinggi berikutnya yaitu R2. R2 dan R5 mempunyai skor yang hampir sama dan dapat dikelompok pada resiko dengan prioritas medium. Dua resiko lainnya, R1 dan R6 mempunyai nilai exposure yang rendah sehingga dapat dikelompokkan pada prioritas rendah.

Dalam kenyataannya, secara umum ada beberapa faktor lain, selain nilai risk exposure, yang harus diperhitungkan pada saat menentukan prioritas resiko.

**Kepercayaan terhadap penilaian resiko**

Beberapa penilaian risk exposure relatif kurang. Untuk idiperlukan investigasi lebih lanjut sebelum tindakan diambil.

**Penggabungan resiko**

Beberapa resiko saling bergantung dengan lainnya. Dalam hal ini maka beberapa resiko tersebut perlu dianggap sebagai satu resiko.

**Jumlah resiko**

Perlu batas jumlah resiko yang dapat dipertimbangkan secara efektif dan dapat diambil tindakannya oleh seorang manajer proyek. Untuk itu perlu dibatasi ukuran daftar prioritas.

**Biaya tindakan**

Beberapa resiko, yang suatu saat dapat dikenali, dapat dikurangi atau dicegah segera dengan biaya atau usaha yang sedikit tanpa menganggap nilai resikonya. Untuk resiko lainnya perlu dilakukan perbandingan antara biaya yang diperlukan dengan benefit yang diperoleh dengan mengurangi resiko tersebut. Satu metode untuk melaksanakan perhitungan ini adalah dengan menghitung risk reduction leverage (RRL) dengan mempergunakan persamaan sebagai berikut:

$$RRL = \frac{RE_{before} - RE_{after}}{\text{Risk reduction cost}}$$

RE<sub>before</sub> adalah nilai risk exposure semula, RE<sub>after</sub> adalah nilai risk exposure yang diharapkan setelah diambil tindakan dan risk education cost adalah biaya untuk implementasi tindakan pengurangan resiko. Risk reduction cost harus dinyatakan dengan unit yang sama dengan nilai resiko yaitu nilai moneter yang diperlukan atau dengan nilai skor. Jika nilai yang diharapkan ternyata lebih besar maka RRL yang lebih besar memperlihatkan bahwa kita perlu berharap untuk meningkatkan rencana pengurangan resiko disebabkan reduksi risk exposure yang diharapkan lebih besar dibandingkan dengan biaya rencana.

**6. MENGURANGI RESIKO**

Ada 5 strategi untuk mengurangi resiko.

**Hazard prevention**

Beberapa bahaya yang akan terjadi dapat dicegah atau dikurangi tingkat resikonya sampai level yang tidak membahayakan. Misal, resiko staf kunci tidak dapat menghadiri rapat dapat diminimisasi dengan membuat jadwal yang lebih awal.

**Likelihood reduction**

Beberapa resiko yang tidak dihindari, dapat dikurangi likelihoodnya dengan membuat rencana prioritas. Misal, resiko terlambatnya perubahan spesifikasi kebutuhan dapat dikurangi melalui prototipe. Melalui prototipe ini tidak akan menghilangkan resiko keterlambatan perubahan dan resiko ini perlu ditambahkan pada rencana kontingensi.

**Risk avoidance**

Misal, sebuah proyek dapat diproteksi dari resiko melewati jadwal yang sudah ditentukan dengan menambah estimasi durasinya atau dengan mengurangi fungsionalitasnya.

**Risk transfer**

Pengaruh dari beberapa resiko dapat dipindahkan dari proyek, misalkan dengan disubkontrakkan.

**Contingency planning**

Pada beberapa resiko yang tidak dapat dicegah maka diperlukan rencana kontingensi untuk mengurangi dampak bahaya yang akan terjadi. Misal, manajer proyek sebaiknya membuat rencana kontingensi untuk mempergunakan programer kontrak untuk meminimisasi dampak akibat ketidakhadiran staf programer yang tidak terencana.

**Tabel 2. Resiko proyek software dan strategi mengurangi resiko**

Resiko	Teknik mengurangi resiko
Kegagalan pada personil	<ul style="list-style-type: none"> <li>• Mempekerjakan staf yang handal</li> <li>• Job matching</li> <li>• Membangun tim</li> <li>• Mengadakan pelatihan dan peningkatan karir</li> <li>• Membuat jadwal lebih awal bagi personil utama</li> </ul>
Estimasi biaya dan waktu yang tidak realistis	<ul style="list-style-type: none"> <li>• Membuat beberapa estimasi</li> <li>• Desain untuk biaya</li> <li>• Meningkatkan pengembangan</li> <li>• Merekam dan menganalisa proyek sebelumnya</li> <li>• Standarisasi metode</li> </ul>
Mengembangkan fungsi software yang salah	<ul style="list-style-type: none"> <li>• Evaluasi proyek ditingkatkan</li> <li>• Buat metode spesifikasi yang formal</li> <li>• Survey pengguna</li> <li>• Buat prototipe</li> <li>• Buat user manual lebih awal.</li> </ul>
Mengembangkan	<ul style="list-style-type: none"> <li>• Membuat prototipe</li> </ul>

an antarmuka pengguna yang salah	<ul style="list-style-type: none"> <li>• Analisis tugas</li> <li>• Keterlibatan pengguna</li> </ul>
Gold plating	<ul style="list-style-type: none"> <li>• Mengurangi kebutuhan</li> <li>• Membuat prototipe</li> <li>• Analisis biaya-manfaat</li> <li>• Desain biaya</li> </ul>
Terlambat untuk mengubah kebutuhan	<ul style="list-style-type: none"> <li>• Mengubah prosedur kendali</li> <li>• Membatasi perubahan yang terlalu banyak</li> <li>• Meningkatkan prototipe</li> <li>• Meningkatkan pengembangan (akibat perubahan)</li> </ul>
Kegagalan pada komponen yang disuplai pihak eksternal	<ul style="list-style-type: none"> <li>• Melakukan benchmarking</li> <li>• Inspeksi</li> <li>• Spesifikasi formal</li> <li>• Kontrak Perjanjian</li> <li>• Prosedur dan sertifikasi jaminan kualitas</li> </ul>
Kegagalan menjalankan tugas eksternal	<ul style="list-style-type: none"> <li>• Prosedur jaminan kualitas</li> <li>• Desain atau prortotipe yang kompetitif</li> <li>• Membangun tim</li> <li>• Kontrak insentif</li> </ul>
Kegagalan kinerja real-time	<ul style="list-style-type: none"> <li>• Simulasi</li> <li>• Benchmarking</li> <li>• Prototipe</li> <li>• Tuning</li> <li>• Analisis teknis</li> </ul>
Pengembangannya terlalu sulit secara teknis	<ul style="list-style-type: none"> <li>• Analisis teknis</li> <li>• Analisis biaya manfaat</li> <li>• Prototype</li> <li>• Melatih dan mengembangkan staf</li> </ul>

## 7. EVALUASI RESIKO TERHADAP JADWAL

Seperti telah kita ketahui bahwa tidak semua resiko dapat dihilangkan – beberapa resiko diklasifikasikan sebagai resiko yang dapat dicegah atau resiko yang dapat dikelola. Dalam hal ini resiko tersebut masih dapat mengakibatkan problem yang berpengaruh terhadap durasi aktifitas. Dengan mengidentifikasi dan mengkategorikan resiko tersebut maka akan terlihat pengaruh resiko tersebut terhadap durasi aktifitas yang sudah direncanakan. Dengan demikian kita akan dapat melakukan evaluasi pengaruh resiko tersebut terhadap rencana aktifitas.

Berikutnya akan dibahas dua metode untuk evaluasi pengaruh ketidakpastian ini terhadap jadwal proyek.

### 7.1. Penggunaan PERT untuk evaluasi pengaruh ketidakpastian

PERT dikembangkan untuk menghitung estimasi ketidakpastian lingkungan terhadap durasi pekerjaan. PERT dikembangkan pada suatu lingkungan proyek yang mahal, beresiko tinggi dan kompleks.

Metode PERT ini memerlukan tiga estimasi :

- **Most likely time**  
Waktu yang diperlukan untuk menyelesaikan pekerjaan dalam situasi normal dan diberikan simbol *m*
- **Optimistic time**  
Waktu tersingkat yang diperlukan untuk menyelesaikan pekerjaan dan diberi simbol *a*.

- **Pessimistic time**

Waktu terlama yang diperlukan untuk menyelesaikan pekerjaan dikarenakan berbagai kemungkinan yang masuk akal dan diberikan simbol *b*.

PERT mengkombinasikan ketiga estimasi tersebut untuk membentuk durasi tunggal yang diharapkan, *t<sub>e</sub>*, dengan mempergunakan rumus:

$$t_e = a + \frac{4m + b}{6}$$

### 7.2. Penggunaan durasi yang diharapkan.

Durasi yang diharapkan dipergunakan supaya suatu forward pass dapat melalui sebuah jaringan; dengan mempergunakan metode yang sama dengan teknik CPM. Akan tetapi dalam hal ini, tanggal aktifitas yang dihitung bukan merupakan tanggal paling awal akan tetapi merupakan tanggal yang diharapkan dapat mencapai aktifitas tersebut.

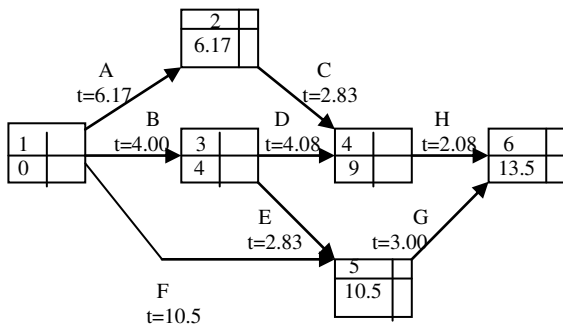
Jaringan PERT yang diperlihatkan pada gambar 3 memperlihatkan bahwa kita berharap proyek tersebut dapat diselesaikan dalam waktu 13,5 minggu- tidak seperti CPM yang tidak memperlihatkan tanggal paling awal untuk menyelesaikan proyek tersebut akan tetapi tanggal yang diharapkan (atau most likely). Salah satu keuntungan dari pendekatan ini adalah menempatkan sebuah emphasis dalam ketidakpastian di dunia nyata.

Tabel 3 berikut ini memperlihatkan contoh estimasi durasi aktifitas yang memperkirakan durasi secara optimistic(*a*), pessimistic(*b*) dan most likely(*m*).

**Tabel 3. Estimasi waktu aktifitas PERT**

Aktifitas	Durasi Aktifitas (minggu)		
	Optimistic (a)	Most Likely (m)	Pessimistic (b)
A	5	6	8
B	3	4	5
C	2	3	3
D	3.5	4	5
E	1	3	4
F	8	10	15
G	2	3	4
H	2	2	2.5

Pendekatan PERT juga difokuskan pada ketidakpastian estimasi durasi aktifitas. Perlu tiga estimasi untuk masing-masing aktifitas yang memperlihatkan fakta bahwa kita tidak yakin dengan apa yang akan terjadi – kita dipaksa untuk menghitung fakta yang diperkirakan akan terjadi.



Gambar 3. Jaringan PERT setelah forward pass

7.3. Deviasi Standar Aktifitas

Perhitungan kuantitatif tingkat ketidakpastian suatu estimasi durasi aktifitas bisa diperoleh dengan menghitung standar deviasi s dari sebuah durasi aktifitas dengan mempergunakan rumus:

$$s = \frac{b - a}{6}$$

Standar deviasi aktifitas proporsional dengan beda antara estimasi optimistic dan pessimistic, dan dapat dipergunakan sebagai tingkatan ukuran level ketidakpastian atau resiko masing-masing aktifitas. Durasi yang diharapkan dari masing-masing aktifitas dan standar deviasi dari proyek tersebut (tabel 3) dapat dilihat pada tabel 4.

Tabel 4. Waktu yang diharapkan dan standar deviasi

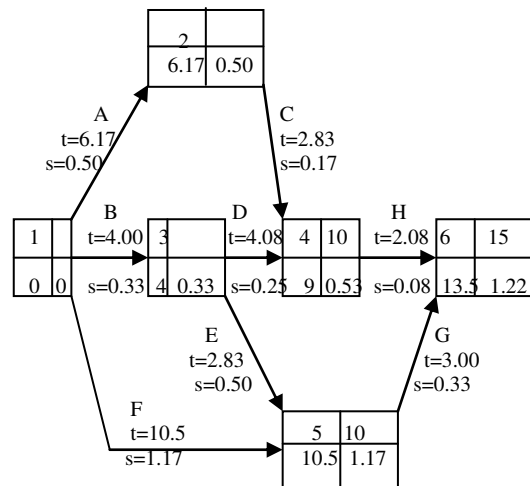
Aktifitas	Durasi Aktifitas (minggu)				
	a	M	b	t <sub>e</sub>	s
A	5	6	8	6.17	0.50
B	3	4	5	4.00	0.33
C	2	3	3	2.83	0.17
D	3.5	4	5	4.08	0.25
E	1	3	4	2.83	0.50
F	8	10	15	10.50	1.17
G	2	3	4	3.00	0.33
H	2	2	2.5	2.08	0.08

a: optimistic    b: most likely    c: pessimistic  
t<sub>e</sub>: expected        s: standard deviation

7.4. Likelihood target terpenuhi

Keuntungan utama dari teknik PERT adalah memberikan suatu metode untuk melakukan estimasi probabilitas tanggal target terpenuhi atau tidak. Teknik ini bisa saja hanya mempunyai tanggal target tunggal yaitu proyek selesai, akan tetapi kita diharapkan untuk mengatur tambahan target antara.

Misalkan kita harus menyelesaikan proyek dalam waktu 15 minggu. Kita berharap proyek tersebut dapat diselesaikan dalam waktu 13.5 minggu akan tetapi durasinya bisa lebih dan bisa kurang.



Gambar 4. Jaringan PERT dengan tiga buah tanggal target dan perhitungan standar deviasi kejadian

Misalkan aktifitas C harus diselesaikan pada minggu ke 10 karena salah satu anggota yang melaksanakan aktifitas tersebut sudah dijadwalkan untuk bekerja pada proyek lain dan kejadian 5 memperlihatkan penyerahan produk kepada pelanggan. Untuk itu diperlukan tiga tanggal target pada jaringan PERT seperti yang diperlihatkan dalam gambar 4.

8. MENGHITUNG NILAI Z

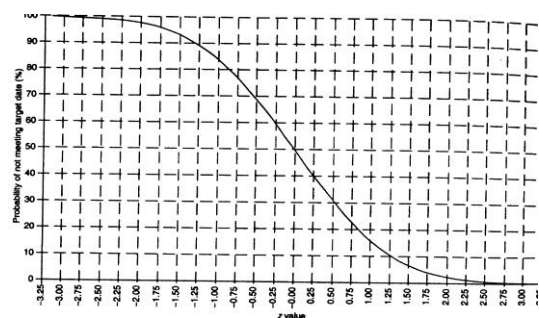
Nilai z dihitung pada setiap node yang mempunyai sebuah tanggal target. Nilai z ini ekuivalen dengan jumlah standar deviasi antara node yang diharapkan dan tanggal target. Nilai z ini dihitung dengan rumus :

$$Z = \frac{T - t_e}{s}$$

t<sub>e</sub> : tanggal yang diharapkan  
T : tanggal target

8.1. Konversi nilai z menjadi probabilitas

Nilai z dapat dikonversikan menjadi probabilitas tidak terpenuhinya tanggal target dengan mempergunakan grafik pada gambar 5.

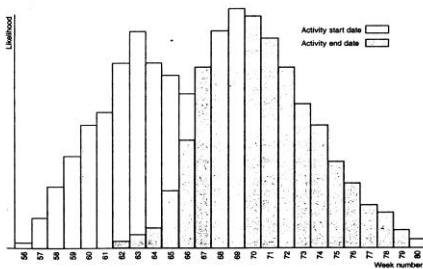


Gambar 5. Probabilitas mendapatkan sebuah nilai dalam standar deviasi z pada mean sebuah distribusi normal

**8.2. Keuntungan PERT**

Kita telah melihat bahwa dengan meminta estimasi durasi aktifitas bernilai banyak dan perhitungan tanggal yang diharapkan, PERT fokus pada ketidakpastian peramalan. Kita dapat mempergunakan teknik tersebut untuk menghitung standar deviasi masing-masing aktifitas dan mempergunakan nilainya untuk membuat urutan tingkatan resiko. Dengan mempergunakan urutan ini, kita dapat melihat, misal, aktifitas F merupakan salah satu aktifitas yang mempunyai ketidakpastian yang terbesar, sedangkan C pada prinsipnya akan memberikan dampak yang kecil.

Jika kita mempergunakan waktu yang diharapkan dan standar deviasi untuk forward passes melalui jaringan yang kita dapat maka untuk sebarang kejadian atau selesainya aktifitas, estimasi probabilitas terpenuhinya target dapat terlaksana. Dalam hal tertentu, dengan mengatur tanggal target sepanjang lintasan kritis, kita dapat fokus pada aktifitas tertentu yang mempunyai resiko yang terbesar terhadap jadwal proyek.



**Gambar 6. Profil resiko pada sebuah aktifitas yang dihasilkan dengan mempergunakan simulasi Monte Carlo.**

Estimasi durasi aktifitas dapat dinyatakan melalui berbagai format bergantung pada informasi

yang diperlukan. Misal jika kita mempunyai data historis mengenai durasi aktifitas serupa, kita bisa saja menyatakan durasi sebagai distribusi probabilitas. Dengan sedikitnya informasi yang kita peroleh, paling tidak kita mampu untuk menghasilkan tiga estimasi waktu sebagai mana yang dipergunakan oleh PERT.

**9. KESIMPULAN**

Keberhasilan proyek perangkat lunak juga dipengaruhi oleh ketepatan melakukan identifikasi resiko dan pengelolaan resiko. Manajemen resiko berkaitan dengan evaluasi dan prioritas resiko serta rencana yang akan dijalankan untuk mengatasi resiko sebelum resiko tersebut menjadi permasalahan.

Tahapan mengelola resiko dimulai dari identifikasi resiko. Selanjutnya resiko tersebut dianalisa untuk menentukan pengaruh dan dampak yang akan terjadi jika resiko tersebut terjadi. Berikutnya adalah menentukan prioritas resiko tersebut berdasar probabilitas dan dampaknya. Terakhir perlu dikembangkan rencana untuk mengelola resiko yang probabilitas terjadinya tinggi dan mempunyai dampak yang besar.

Beberapa resiko yang dapat berpengaruh terhadap proyek perangkat lunak dapat direduksi dengan menempatkan beberapa staf yang sudah profesional pada beberapa aktifitas yang terpengaruh resiko tersebut.

**10. DAFTAR PUSTAKA**

1. Bob Hughes, Mike Cotterell, "Software Project Management (Second Edition)", McGraw-Hill, 1999.
2. Roger S. Pressman, "Software Engineering", McGraw-Hill, 2000.