

**MANAJEMEN RISIKO PADA
PROYEK PEMBUATAN PERANGKAT LUNAK
“Giftiesiti”**



LAPORAN

**Disusun untuk Memenuhi Tugas Individu
pada Mata Kuliah Manajemen Proyek Semester Enam
yang Diampu oleh Dinar Mutiara K. N., S.T., M.Info., Ph.D.**

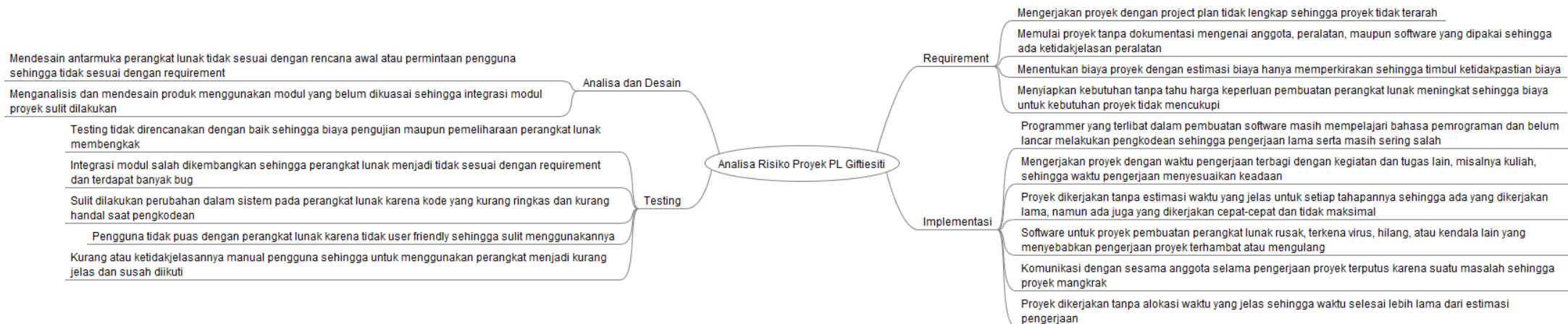
OLEH :

**Anjar Asmarani
NIM. 24060119130095**

**PROGRAM STUDI S-1 INFORMATIKA KELAS B
FAKULTAS SAINS DAN MATEMATIKA
UNIVERSITAS DIPONEGORO
SEMARANG
2022**

A. Identifikasi Risiko

Risiko dalam proyek pembuatan perangkat lunak Giftiesiti dibagi ke dalam empat bagian yang akan diidentifikasi, yaitu identifikasi risiko yang muncul untuk tahap *requirement*, tahap analisa dan desain, tahap implementasi, serta tahap *testing*. Identifikasi dilakukan dengan mengidentifikasi risiko yang muncul dengan struktur pendefinisian <1.kondisi terkini>, <2.kejadian tak terduga> dapat menyebabkan <3. dampak pada pencapaian sasaran>. Identifikasi risiko ini dibuat dengan *brain storming* dan disajikan dalam bentuk *mind mapping* seperti pada gambar berikut.



B. Mengases/Menilai Risiko

Dalam mengases atau menilai risiko, didasarkan pada matriks berikut.

MENGASES /MENILAI RISIKO

LIKELIHOOD	5	Sedang (5x1)	Tinggi (5x2)	Ekstrim (5x3)	Ekstrim (5x4)	Ekstrim (5x5)
	4	Sedang (4x1)	Tinggi (4x2)	Ekstrim (4x3)	Ekstrim (4x4)	Ekstrim (4x5)
	3	Rendah (3x1)	Sedang (3x2)	Tinggi (3x3)	Ekstrim (3x4)	Ekstrim (3x5)
	2	Rendah (2x1)	Sedang (2x2)	Sedang (2x3)	Tinggi (2x4)	Tinggi (2x5)
	1	Rendah (1x1)	Rendah (1x2)	Rendah (1x3)	Sedang (1x4)	Sedang (1x5)
		1	2	3	4	5
DAMPAK/KONSEKWENSI						

1-3
4-6
7-10
11-25

• RENDAH
• SEDANG
• TINGGI
• EKSTRIM

Matriks tersebut membagi risiko ke dalam empat kelompok jenis risiko berdasarkan *likelihood* dan dampaknya. Nilai *likelihood* dikalikan dengan nilai dampak untuk menghasilkan sebuah nilai yang menentukan risiko tersebut masuk ke dalam kelompok yang mana di antara kelompok “rendah”, “sedang”, “tinggi”, atau kelompok “ekstrim”.

Penilaian risiko untuk proyek pembuatan perangkat lunak Giftiesiti disajikan dalam tabel di bawah ini.

No.	Kejadian	Likelihood	Dampak	Jenis Risiko
1.	Requirement			
a.	Mengerjakan proyek dengan <i>project plan</i> tidak lengkap sehingga proyek tidak terarah	Rendah	Tinggi	Tinggi
b.	Memulai proyek tanpa dokumentasi mengenai anggota, peralatan, maupun <i>software</i> yang dipakai sehingga ada ketidakjelasan peralatan	Sedang	Rendah	Rendah
c.	Menentukan biaya proyek dengan estimasi biaya hanya memperkirakan sehingga timbul ketidakpastian biaya	Rendah	Rendah	Rendah

d.	Menyiapkan kebutuhan tanpa tahu harga keperluan pembuatan perangkat lunak meningkat sehingga biaya untuk kebutuhan proyek tidak mencukupi	Rendah	Rendah	Rendah
2.	Analisis dan Desain			
a.	Mendesain antarmuka perangkat lunak tidak sesuai dengan rencana awal atau permintaan pengguna sehingga tidak sesuai dengan <i>requirement</i>	Rendah	Tinggi	Sedang
b.	Menganalisis dan mendesain produk menggunakan modul yang belum dikuasai sehingga integrasi modul proyek sulit dilakukan	Sedang	Ekstrim	Ekstrim
3.	Implementasi			
a.	<i>Programmer</i> yang terlibat dalam pembuatan <i>software</i> masih mempelajari bahasa pemrograman dan belum lancar melakukan pengkodean sehingga pengerjaan lama serta masih sering salah	Sedang	Tinggi	Ekstrim
b.	Mengerjakan proyek dengan waktu pengerjaan terbagi dengan kegiatan dan tugas lain, misalnya kuliah, sehingga waktu pengerjaan menyesuaikan keadaan	Sedang	Rendah	Rendah
c.	Proyek dikerjakan tanpa estimasi waktu yang jelas untuk setiap tahapannya sehingga ada yang dikerjakan lama, namun ada juga yang dikerjakan cepat-cepat dan tidak maksimal	Rendah	Tinggi	Sedang
d.	<i>Software</i> untuk proyek pembuatan perangkat lunak rusak, terkena virus, hilang, atau kendala lain yang	Rendah	Ekstrim	Tinggi

	menyebabkan pengerjaan proyek terhambat atau mengulang			
e.	Komunikasi dengan sesama anggota selama pengerjaan proyek terputus karena suatu masalah sehingga proyek mangkrak	Rendah	Tinggi	Sedang
f.	Proyek dikerjakan tanpa alokasi waktu yang jelas sehingga waktu selesai lebih lama dari estimasi pengerjaan	Rendah	Tinggi	Sedang
4.	Testing			
a.	<i>Testing</i> tidak direncanakan dengan baik sehingga biaya pengujian maupun pemeliharaan perangkat lunak membengkak	Rendah	Sedang	Rendah
b.	Integrasi modul salah dikembangkan sehingga perangkat lunak menjadi tidak sesuai dengan <i>requirement</i> dan terdapat banyak <i>bug</i>	Sedang	Ekstrem	Ekstrem
c.	Sulit dilakukan perubahan dalam sistem pada perangkat lunak karena kode yang kurang ringkas dan kurang handal saat pengkodean	Rendah	Sedang	Sedang
d.	Pengguna tidak puas dengan perangkat lunak karena tidak <i>user friendly</i> sehingga sulit menggunakannya	Rendah	Sedang	Sedang
e.	Kurang atau ketidakjelasan manual pengguna sehingga untuk menggunakan perangkat menjadi kurang jelas dan susah diikuti	Rendah	Tinggi	Tinggi

C. Respon terhadap Risiko

Respon untuk risiko yang telah diidentifikasi dan dinilai dapat berupa *avoid*, *mitigate*, *transfer*, maupun *accept*. Untuk respon terhadap risiko yang telah diidentifikasi dan dinilai pada proyek pembuatan perangkat lunak Giftiesiti, disajikan dalam tabel berikut.

No.	Kejadian	Jenis Risiko	Respon
1.	Requirement		
a.	Mengerjakan proyek dengan <i>project plan</i> tidak lengkap sehingga proyek tidak terarah	Tinggi	<i>Avoid</i> , karena <i>project plan</i> berkaitan erat dengan proyek yang sedang dijalankan
b.	Memulai proyek tanpa dokumentasi mengenai anggota, peralatan, maupun <i>software</i> yang dipakai sehingga ada ketidakjelasan peralatan	Rendah	<i>Avoid</i> , karena biaya dokumentasi terjangkau dan jika tidak ada dokumentasi akan membingungkan
c.	Menentukan biaya proyek dengan estimasi biaya hanya memperkirakan sehingga timbul ketidakpastian biaya	Rendah	<i>Mitigate</i> , dengan cara membandingkan jumlah hari yang diperkirakan untuk penyelesaian proyek
d.	Menyiapkan kebutuhan tanpa tahu harga keperluan pembuatan perangkat lunak meningkat sehingga biaya untuk kebutuhan proyek tidak mencukupi	Rendah	<i>Mitigate</i> , dengan cara mencari tahu harga keperluan guna membuat perangkat lunak yang akan dikerjakan
2.	Analisis dan Desain		
a.	Mendesain antarmuka perangkat lunak tidak sesuai dengan rencana awal atau permintaan pengguna sehingga tidak sesuai dengan <i>requirement</i>	Sedang	<i>Avoid</i> , karena telah ada rencana dan kesepakatan atau permintaan atas proyek yang dapat menuntun pembuatan antarmuka yang sesuai
b.	Menganalisis dan mendesain produk menggunakan modul yang belum dikuasai sehingga integrasi modul proyek sulit dilakukan	Ekstrim	<i>Accept</i> , karena modul telah ditentukan sesuai permintaan klien sehingga <i>programmer</i> harus mempelajari modul yang digunakan

3. Implementasi			
a.	<i>Programmer</i> yang terlibat dalam pembuatan <i>software</i> masih mempelajari bahasa pemrograman dan belum lancar melakukan pengkodean sehingga pengerjaan lama serta masih sering salah	Ekstrim	<i>Accept</i> , karena bahasa pemrograman telah ditentukan sesuai permintaan klien sehingga <i>programmer</i> harus mempelajari cara pengkodean yang digunakan
b.	Mengerjakan proyek dengan waktu pengerjaan terbagi dengan kegiatan dan tugas lain, misalnya kuliah, sehingga waktu pengerjaan menyesuaikan keadaan	Rendah	<i>Mitigate</i> , dengan cara membuat jadwal dan pembagian waktu untuk pengerjaan proyek dan aktivitas atau kegiatan lain
c.	Proyek dikerjakan tanpa estimasi waktu yang jelas untuk setiap tahapannya sehingga ada yang dikerjakan lama, namun ada juga yang dikerjakan cepat-cepat dan tidak maksimal	Sedang	<i>Avoid</i> , karena estimasi waktu dapat dibuat dengan melakukan penjadwalan dan menargetkan setiap tahapan untuk waktu tertentu dan dapat dijadikan pedoman pengerjaan
d.	<i>Software</i> untuk proyek pembuatan perangkat lunak rusak, terkena virus, hilang, atau kendala lain yang menyebabkan pengerjaan proyek terhambat atau mengulang	Tinggi	<i>Avoid</i> , karena masalah pada <i>software</i> dapat diantisipasi dengan menyiapkan <i>software</i> yang mumpuni, memasang antivirus, serta menjaga dan memeliharanya sebaik mungkin
e.	Komunikasi dengan sesama anggota selama pengerjaan proyek terputus karena suatu masalah sehingga proyek mangkrak	Sedang	<i>Mitigate</i> , dengan cara membuat <i>backup plan</i> jika salah satu anggota tiba-tiba tidak bisa dihubungi
f.	Proyek dikerjakan tanpa alokasi waktu yang jelas sehingga waktu selesai lebih lama dari estimasi pengerjaan	Sedang	<i>Avoid</i> , karena waktu pengerjaan dapat direncanakan sebelum memulai implementasi proyek
4. Testing			
a.	<i>Testing</i> tidak direncanakan dengan baik sehingga biaya pengujian maupun pemeliharaan perangkat lunak	Rendah	<i>Avoid</i> , karena rencana pengujian dan pemeliharaan harus sudah dipikirkan sejak tahapan awal pelaksanaan

	membengkok		proyek
b.	Integrasi modul salah dikembangkan sehingga perangkat lunak menjadi tidak sesuai dengan <i>requirement</i> dan terdapat banyak <i>bug</i>	Ekstrim	<i>Avoid</i> , karena sebelum tahap pengujian harus dipastikan bahwa integrasi modul telah sesuai dengan <i>requirement</i>
c.	Sulit dilakukan perubahan dalam sistem pada perangkat lunak karena kode yang kurang ringkas dan kurang handal saat pengkodean	Sedang	<i>Mitigate</i> , dengan cara memperbaiki kode atau meringkas dan memperjelasnya agar mudah untuk dilakukan perubahan maupun pengembangan
d.	Pengguna tidak puas dengan perangkat lunak karena tidak <i>user friendly</i> sehingga sulit menggunakannya	Sedang	<i>Accept</i> , karena penilaian pengguna tidak dapat diganggu gugat dan tergantung dengan individu yang menilai
e.	Kurang atau ketidakjelasan manual pengguna sehingga untuk menggunakan perangkat menjadi kurang jelas dan susah diikuti	Tinggi	<i>Mitigate</i> , dengan membuat atau memperbaiki manual dengan bahasa yang jelas dan mudah dimengerti serta dapat dipahami oleh pengguna