BAB 2
TINJAUAN PUSTAKA

2.1 Profil PT. Anantagraha Primaperkasa

Pada bagian ini akan dijelaskan mengenai tempat penelitian yaitu PT. Anantagraha Primaperkasa.

2.1.1 Sejarah PT. Anantagraha Primaperkasa

PT. Anantagraha Primaperkasa merupakan sebuah perusahaan yang bergerak dalam bidang jasa konstruksi. Perusahaan ini didirikan oleh Bapak Ir. Budianto Azis, M.B.A. sejak tahun 1994 dan beliau merupakan direktur perusahaan tersebut. Hingga saat ini PT. Anantagraha Primaperkasa telah mengerjakan berbagai proyek tender antara lain pabrik, showroom, rumah sakit, apartemen, hotel, bangunan komersial, dan sekolah.

2.1.2 Logo PT. Anantagraha Primaperkasa

Berikut ini merupakan logo dari PT. Anantagraha Primaperkasa yang bisa dilihat pada Gambar 2.1.



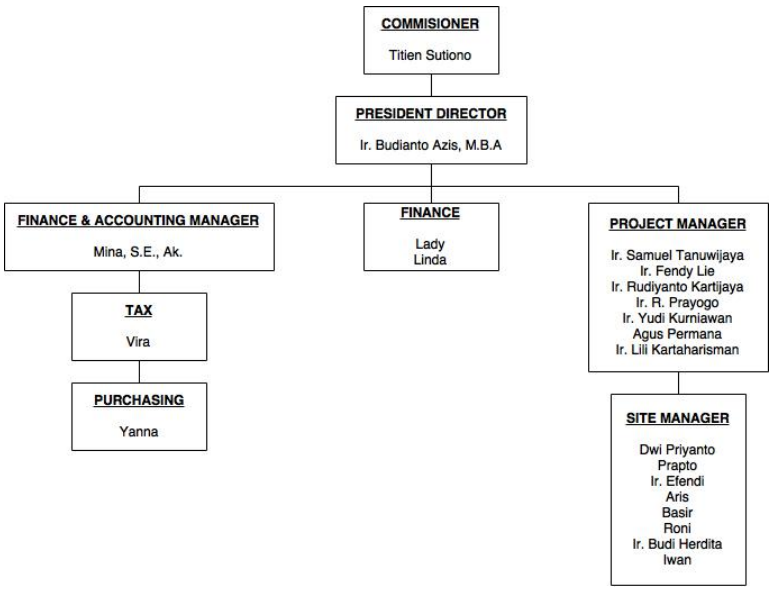
Gambar 2.1 Logo PT. Anantagraha Primaperkasa

2.1.3 Badan Hukum PT. Anantagraha Primaperkasa

PT. Anantagraha Primaperkasa sudah memiliki izin usaha dengan nomor Surat Izin Usaha Perdagangan (SIUP) 10165-03/PM/1.824.271 yang terdaftar di Kota Jakarta Barat

2.1.4 Struktur Organisasi

Untuk menjelaskan struktur organisasi yang menjadi sasaran kegiatan penelitian maka struktur organisasi dari PT. Anantagraha Primaperkasa dapat dilihat pada Gambar 2.2.



Gambar 2.2 Struktur Organisasi

2.1.5 Deskripsi Kerja Jabatan

Berikut adalah pemaparan deskripsi kerja (*job description*) untuk tiap-tiap jabatan di PT. Anantagraha Primaperkasa:

1. Commisioner
    a. Memilih, menentukan dan mengawasi direktur
    b. Mengawasi jalannya perusahaan
    c. Memberi masukkan bagi perusahaan
2. President Director
    a. Memimpin perusahaan dengan membuat kebijakan-kebijakan perusahaan
    b. Memilih, menentukan, dan mengawasi pekerjaan karyawan
    c. Menyetujui anggaran tahunan perusahaan
3. Finance & Accounting Manager
    a. Membuat laporan aktivitas keuangan perusahaan
    b. Membuat laporan pajak perusahaan
    c. Membuat anggaran pengeluaran perusahaan
    d. Membuat anggaran pendapatan perusahaan
    e. Membuat dokumen tagihan dan tanda terima keuangan

4. Tax
    a. Mengelola audit pajak
    b. Investigasi perkembangan undang-undang pajak baru
    c. Identifikasi masalah yang mungkin timbul dalam transaksi pajak
    d. Mengorganisir ketentuan pajak penghasilan
    e. Menghitung pembayaran pajak penghasilan
5. Purchasing
    a. Membuat laporan pembelian dan pengeluaran barang
    b. Memastikan kesediaan barang
    c. Melakukan pembayaran kepada pemasok
6. Finance
    a. Mengatur keuangan perusahaan
    b. Melakukan transaksi keuangan perusahaan
    c. Menagih pembayaran kepada pelanggan
    d. Mengatur aktivitas keuangan perusahaan
    e. Melakukan evaluasi anggaran
7. Project Manager
    a. Memeriksa estimasi RAB yang dibuat dan menyetujui penjadwalan yang dibuat oleh *Site Manager*
    b. Melakukan negosiasi dengan pemasok dan mandor
    c. Mengamati pengeluaran biaya proyek
    d. Berhubungan dengan pihak ketiga (owner dan kontraktor lain)
    e. Membuat perencanaan kegiatan operasional pelaksanaan proyek
    f. Mengatur kegiatan operasional pelaksanaan proyek
    g. Melaksanakan kegiatan operasional pelaksanaan proyek
    h. Mengendalikan pelaksanaan operasional pelaksanaan proyek
8. Site Manager
    a. Menerima tugas dari manajer proyek untuk mengatur jalannya proyek
    b. Mengatur pelaksanaan gambar dan spesifikasi teknis
    c. Menerima gambar desain dan spesifikasi teknis proyek dari manajer proyek dan mempelajarinya

d. Mengatur pelaksanaan penjadwalan dan metode pelaksanaan pekerjaan
e. Mengatur tenaga kerja dan sub-kontraktor
f. Mengatur pengadaan material di lapangan
g. Bertanggung jawab sepenuhnya kepada manajer proyek atas keberhasilan pelaksanaan proyek yang dipegangnya

2.2 Landasan Teori

Untuk membangun Sistem Informasi Manajemen Risiko Proyek di PT. Anantagraha Primaperkasa diperlukan teori-teori yang mendukung serta berkaitan dengan perangkat lunak yang akan dikembangkan. Teori-teori tersebut dapat menjadi landasan dari perangkat lunak yang akan dikembangkan.

2.2.1 Sistem Informasi

Sistem adalah sekumpulan komponen yang saling bekerja sama untuk mencapai suatu tujuan. Masing-masing komponen memiliki fungsi yang berbeda dengan yang lain, tetapi tetap dapat bekerja sama. Fungsi sistem yang utama adalah menerima masukkan, mengolah masukkan dan menghasilkan keluaran. Agar dapat menjalankan fungsinya ini, sistem akan memiliki komponen-komponen masukkan, proses, keluaran, dan kontrol untuk menjamin bahwa semua fungsi dapat berjalan dengan baik.

Informasi adalah data yang sudah diolah sehingga berguna untuk pembuatan keputusan. Data adalah representasi suatu objek. Misalnya seorang mahasiswa diwakili oleh nomor mahasiswa, maka nomor mahasiswa ini adalah data. Data yang belum diolah belum dapat digunakan untuk pembuatan keputusan.

Manajemen adalah sekumpulan orang yang bertugas menjalankan perusahaan, organisasi, atau instansi pemerintah. Manajemen bertugas menentukan tujuan perusahaan, merencanakan kegiatan untuk satu periode mendatang, dan menjalankan rencana tersebut dengan melibatkan seluruh pihak terkait di dalam perusahaan.

Apabila masing-masing pengertian di atas digabung, akan diperoleh pengertian sistem informasi adalah sekumpulan komponen yang saling bekerja sama, yang digunakan untuk mencatat data, mengolah data dan menyajikan

informasi untuk para pembuat keputusan agar dapat membuat keputusan dengan baik.

Komponen sistem informasi sangat tergantung kepada proses yang terjadi di masing-masing perusahaan. Komponen yang paling utama adalah teknologi komunikasi, teknologi komputasi, dan teknologi informasi. Teknologi komunikasi digunakan untuk mengirim data dari suatu tempat atau alat ke tempat atau alat yang lain. Teknologi komputasi adalah berbagai perangkat yang digunakan untuk mengolah data. Teknologi informasi adalah berbagai metode untuk menyajikan berbagai bentuk informasi ke berbagai pihak yang memerlukannya [1].

2.2.2 Proyek

Proyek adalah upaya sementara yang dilakukan untuk menciptakan produk, layanan, atau hasil yang unik. Arti kata sementara yaitu memiliki awal dan akhir yang pasti. Akan tetapi produk, layanan, atau hasil yang dibuat oleh sebuah proyek bersifat jangka panjang. Sebuah proyek dapat dikatakan mencapai akhir apabila tujuan dari proyek tersebut telah terpenuhi, atau sudah tidak ada lagi kebutuhan dari proyek. Proyek juga dapat dihentikan jika pelanggan memang ingin menghentikan proyek. Setiap proyek mencitakan produk, layanan, atau hasil yang unik. Hasil dari proyek tersebut dapat berwujud atau tidak berwujud. Arti kata unik yaitu memiliki karakteristik yang berbeda, misal dalam segi bahan, lokasi, desain, situasi, *stakeholder*, dan sebagainya [2].

2.2.3 Manajemen Proyek

Manajemen proyek adalah penerapan pengetahuan, keterampilan, peralatan, dan teknik dalam kegiatan proyek untuk memenuhi kebutuhan proyek. Manajemen proyek terdiri dari 5 kelompok proses, yaitu memulai (*initiating*), merencanakan (*planning*), melaksanakan (*executing*), mengawasi dan mengendalikan (*monitoring and controlling*), dan menutup (*closing*). Manajemen proyek terdiri dari 10 *knowledge area*, yaitu [2]:

1. *Project Integration Management*

   Manajemen integrasi proyek mencakup proses dan aktivitas untuk mengidentifikasi, menentukan, menggabungkan, menyatukan, dan mengkoordinasikan berbagai proses dan aktivitas manajemen proyek.

2. *Project Scope Management*

   Manajemen ruang lingkup proyek mencakup proses yang diperlukan untuk memastikan bahwa proyek mencakup semua pekerjaan yang dibutuhkan, dan hanya pekerjaan yang dibutuhkan untuk menyelesaikan proyek dengan sukses.

3. *Project Time Management*

   Manajemen jadwal proyek mencakup proses yang diperlukan untuk mengelola penyelesaian proyek secara tepat waktu.

4. *Project Cost Management*

   Manajemen biaya proyek mencakup proses yang terlibat dalam perencanaan, estimasi, penganggaran, pembiayaan, pendanaan, pengelolaan, dan pengendalian biaya sehingga proyek dapat diselesaikan dalam anggaran yang disetujui.

5. *Project Quality Management*

   Manajemen mutu proyek mencakup proses dan aktivitas organisasi kinerja yang menentukan kebijakan mutu, tujuan, dan tanggung jawab sehingga proyek akan memenuhi kebutuhan yang dimilikinya.

6. *Project Human Resource Management*

   Manajemen sumber daya manusia proyek mencakup proses dan aktivitas organisasi kinerja yang menentukan kebijakan mutu, tujuan, dan tanggung jawab sehingga proyek akan memenuhi kebutuhan yang dimilikinya.

7. *Project Communications Management*

   Manajemen komunikasi proyek mencakup proses yang diperlukan untuk memastikan perencanaan, pengumpulan, pembuatan, pendistribusian, penyimpanan, pengambilan, pengendalian, pemantauan, dan disposisi informasi yang tepat.

8. *Project Risk Management*

   Manajemen risiko proyek mencakup proses dalam melakukan perencanaan manajemen risiko, identifikasi, analisis, perencanaan respons, dan pengendalian risiko pada suatu proyek.

9. *Project Procurement Management*

   Manajemen pengadaan proyek mencakup proses yang diperlukan untuk membeli atau memperoleh produk, layanan, atau hasil yang dibutuhkan dari luar tim proyek.

10. *Project Stakeholder Management*

    Manajemen *stakeholder* proyek mencakup proses yang diperlukan untuk mengidentifikasi orang, kelompok, atau organisasi yang dapat berdampak atau terkena dampak proyek untuk menganalisis harapan *stakeholder* dan dampaknya terhadap proyek, dan untuk mengembangkan strategi manajemen yang tepat untuk melibatkan para *stakeholder* secara efektif dalam keputusan proyek dan eksekusi.

2.2.4 Manajemen Risiko Proyek

Manajemen risiko proyek mencakup proses dalam melakukan perencanaan manajemen risiko, identifikasi risiko, analisis risiko, perencanaan respons terhadap risiko, dan pengendalian risiko pada suatu proyek. Tujuan pengelolaan risiko proyek adalah untuk meningkatkan kemungkinan dan dampak kejadian positif, dan mengurangi kemungkinan dan dampak kejadian negatif dalam proyek [2].

2.2.4.1 Perencanaan Manajemen Risiko

Perencanaan manajemen risiko adalah proses bagaimana melakukan kegiatan pengelolaan risiko untuk suatu proyek. Kunci dari proses ini adalah memastikan bahwa tingkat, jenis, dan visibilitas manajemen risiko sama dengan risiko dan pentingnya proyek terhadap organisasi. Rencana pengelolaan risiko sangat penting untuk dikomunikasikan dan mendapatkan persetujuan dan dukungan dari semua *stakeholder* untuk memastikan agar proses manajemen risiko didukung dan dilakukan secara efektif selama siklus hidup proyek.

Perencanaan yang cermat dan eksplisit meningkatkan kemungkinan keberhasilan bagi proses manajemen risiko lainnya. Perencanaan juga penting untuk menyediakan sumber daya dan waktu yang tepat untuk kegiatan manajemen risiko dan untuk menetapkan dasar yang disepakati untuk mengevaluasi risiko. Proses manajemen risiko harus dimulai saat sebuah proyek disusun dan harus diselesaikan lebih awal selama perencanaan proyek [2].

2.2.4.2 Identifikasi Risiko

Identifikasi risiko adalah sebuah proses untuk menentukan risiko yang dapat mempengaruhi proyek dan mendokumentasikan karakteristik mereka. Kunci dari proses ini adalah dokumentasi risiko dan pengetahuan serta kemampuan yang ada pada tim proyek untuk mengantisipasi kejadian risiko.

Peserta dalam kegiatan identifikasi risiko dapat terdiri dari: manajer proyek, anggota tim proyek, tim manajemen risiko (jika ditugaskan), pelanggan, pakar materi dari luar tim proyek, pengguna akhir, manajer proyek lainnya, *stakeholder*, dan pakar manajemen risiko. Meskipun personil ini sering menjadi peserta kunci untuk identifikasi risiko, semua personil proyek harus didorong untuk mengidentifikasi potensi risiko.

Mengidentifikasi risiko adalah proses yang berulang, karena risiko baru dapat berkembang atau dikenal saat proyek berjalan melalui siklus hidupnya. Frekuensi iterasi dan partisipasi dalam setiap siklus akan bervariasi menurut situasi. Format pernyataan risiko harus konsisten untuk memastikan bahwa setiap risiko dipahami secara jelas dan tidak jelas untuk mendukung analisis dan pengembangan respons yang efektif. Pernyataan risiko harus mendukung kemampuan untuk membandingkan dampak relatif dari satu risiko terhadap orang lain dalam proyek. Prosesnya harus melibatkan tim proyek sehingga mereka dapt mengembangkan dan memelihara rasa memiliki dan bertanggung jawab atas risiko dan tindakan respons risiko terkait. *Stakeholder* di luar tim proyek dapat memberikan informasi obyektif tambahan [2].

2.2.4.3 Analisis Risiko Kualitatif

Analisis risiko kualitatif adalah proses memprioritaskan risiko untuk analisis atau tindakan lebih lanjut dengan menilai dan menggabungkan kemungkinan terjadinya dan dampaknya. Kunci dari proses ini adalah memungkinkan manajer proyek mengurangi tingkat ketidakpastian dan fokus pada prioritas risiko tinggi.

Analisis risiko kualitatif menilai prioritas risiko yang teridentifikasi dengan menggunakan kemungkinan relatif atau kemungkinan terjadinya, dampak yang terkait dengan tujuan proyek jika risiko terjadi, serta faktor lain seperti kerangka waktu untuk respons dan toleransi risiko organisasi yang terkait dengan kendala proyek biaya, jadwal, ruang lingkup, dan kualitas. Penilaian tersebut mencerminkan sikap risiko tim proyek dan *stakeholder* lainnya. Oleh karena itu, penilaian yang efektif memerlukan identifikasi dan pengelolaan eksplisit dari pendekatan risiko peserta kunci dalam proses analisis risiko kualitatif. Dimana pendekatan risiko ini memperkenalkan bias ke dalam penilaian risiko yang teridentifikasi, perhatian harus diberikan untuk mengidentifikasi bias dan koreksi untuk itu.

Menetapkan definisi tingkat probabilitas dan dampak dapat mengurangi pengaruh bias. Kekritisan waktu tindakan terkait risiko dapat memperbesar pentingnya risiko. Evaluasi kualitas informasi yang tersedia mengenai risiko proyek juga membantu mengklarifikasi penilaian kepentingan risiko terhadap proyek.

Analisis risiko kualitatif biasanya merupakan cara yang cepat dan hemat biaya untuk menetapkan prioritas tanggapan respons terhadap risiko dan meletakkan dasar untuk melakukan analisis risiko kuantitatif, jika diperlukan. Analisis risiko kualitatif dilakukan secara rutin sepanjang siklus hidup proyek, seperti yang dijelaskan dalam rencana manajemen risiko proyek. Proses ini dapat mengarah ke analisis risiko kuantitatif atau langsung ke perencanaan respons terhadap risiko [2].

2.2.4.4 Analisis Risiko Kuantitatif

Analisis risiko kuantitatif adalah proses analisis secara numerik pengaruh risiko yang teridentifikasi terhadap keseluruhan tujuan proyek. Kunci dari proses ini adalah menghasilkan informasi risiko kuantitatif untuk mendukung pengambilan keputusan guna mengurangi ketidakpastian proyek.

Analisis risiko kuantitatif dilakukan terhadap risiko yang telah diprioritaskan oleh proses analisis risiko kualitatif yang berpotensi dan secara substansial mempengaruhi tuntutan persaingan proyek. Analisis risiko kuantitatif menganalisis dampak risiko tersebut terhadap tujuan proyek. Hal ini digunakan terutama untuk mengevaluasi efek keseluruhan dari semua risiko yang mempengaruhi proyek. Bila risiko mendorong analisis kuantitatif, proses tersebut dapat digunakan untuk menetapkan peringkat prioritas numerik terhadap risiko tersebut secara individual.

Analisis risiko kuantitatif umumnya mengikuti proses analisis risiko kualitatif. Dalam beberapa kasus, tidak mungkin melakukan proses analisis risiko kuantitatif karena kurangnya data yang memadai untuk mengembangkan model yang sesuai. Manajer proyek harus menggunakan pertimbangan ahli untuk menentukan kebutuhan dan kelangsungan analisis risiko kuantitatif. Ketersediaan waktu dan anggaran, dan kebutuhan akan pernyataan kualitatif atau kuantitatif tentang risiko dan dampak, akan menentukan metode mana yang akan digunakan pada proyek tertentu. Analisis risiko kuantitatif harus diulang, jika diperlukan, sebagai bagian dari proses kontrol risiko untuk menentukan apakah keseluruhan risiko proyek telah menurun secara memuaskan. Tren mungkin menunjukkan kebutuhan untuk fokus sedikit atau lebih pada aktivitas manajemen risiko yang tepat [2].

2.2.4.5 Perencanaan Respons Terhadap Risiko

Perencanaan respons terhadap risiko adalah proses pengembangan opsi dan tindakan untuk meningkatkan peluang dan mengurangi ancaman terhadap tujuan proyek. Kunci utama proses ini adalah menangani risiko berdasarkan

prioritas mereka, memasukkan sumber daya dan aktivitas ke dalam anggaran jadwal dan rencana pengelolaan proyek sesuai kebutuhan.

Perencanaan respons terhadap risiko mengikuti analisis risiko kuantitatif (jika digunakan). Setiap respons risiko memerlukan pemahaman mekanisme yang dengannya akan mengatasi risiko. Ini adalah mekanisme yang digunakan untuk menganalisis apakah rencana respons risiko memiliki efek yang diinginkan. Ini termasuk identifikasi dan penugasan satu orang (pemilik respons risiko) untuk bertanggung jawab atas setiap respons risiko yang disepakati dan didanai. Respons risiko harus sesuai dengan signifikansi risiko, hemat biaya dalam memenuhi tantangan, realistis dalam konteks proyek, disepakati oleh semua pihak yang terlibat, dan dimiliki oleh orang yang bertanggung jawab. Memilih respons risiko yang optimal dari beberapa pilihan sering kali dibutuhkan.

Perencanaan respons terhadap risiko menyajikan pendekatan yang umum digunakan untuk merencanakan respons terhadap risiko. Risiko meliputi ancaman dan peluang yang dapat mempengaruhi keberhasilan proyek dan tangga[an dibahas untuk masing-masing [2].

2.2.4.6 Pengendalian Risiko

Pengendalian risiko adalah proses penerapan perencanaan respons terhadap risiko, melacak risiko yang teridentifikasi, memantau risiko residual, mengidentifikasi risiko baru, dan mengevaluasi keefektifan proses risiko selama proyek berlangsung. Kunci dari proses ini adalah meningkatkan efisiensi pendekatan risiko sepanjang siklus hidup proyek untuk terus mengoptimalkan respons risiko.

Perencanaan respons terhadap risiko yang termasuk dalam daftar risiko dilaksanakan selama siklus hidup proyek, namun pekerjaan proyek harus terus dipantau untuk risiko baru, perubahan, dan usang.

Proses pengendalian risiko menerapkan teknik, seperti varian dan analisis kecenderungan, yang memerlukan penggunaan informasi kinerja yang dihasilkan selama pelaksanaan proyek. Tujuan lain dari proses pengendalian risiko adalah untuk menentukan apakah:

1. Asumsi proyek masih berlaku,
2. Analisis menunjukkan bahwa risiko yang dinilai telah berubah atau dapat dihentikan,
3. Kebijakan dan prosedur manajemen risiko diikuti, dan
4. Cadangan kontingensi untuk biaya atau jadwal harus disesuaikan dengan penilaian risiko saat ini.

Pengendalian risiko dapat melibatkan pemilihan strategi alternatif, melaksanakan rencana kontingensi atau mundur, mengambil tindakan korektif, dan memodifikasi rencana manajemen proyek. Pemilik respons risiko melaporkan secara berkala kepada manajer proyek mengenai keefektifan rencana tersebut, setiap efek yang tidak diantisipasi, dan koreksi yang diperlukan untuk menangani risiko secara tepat. Pengendalian risiko juga mencakup pembaharuan aset organisasi, termasuk pelajaran proyek yang dipelajari *database* dan *template* manajemen risiko, untuk keuntungan proyek masa depan [2].

2.2.5 Penanganan Risiko

Penanganan risiko dimaksudkan agar jenis dan biaya risiko yang nilai nominalnya telah dihitung, dapat dikelola atau ditangani sehingga solusi serta penanggung jawab risikonya dapat ditentukan. Beberapa cara menentukan penanganan risiko berdasarkan klasifikasi bentuk risikonya, yaitu [3]:

1. Risiko yang dapat diterima

   Merupakan bentuk risiko yang ditanggulangi oleh individu/perusahaan karena konsekuensinya dinilai cukup kecil.
2. Risiko yang direduksi

   Merupakan bentuk risiko yang dapat ditangani dengan cara menangani suatu tindakan alternatif yang nilai konsekuensinya dapat saja nihil atau paling tidak konsekuensi yang ditanggung lebih kecil.
3. Risiko yang dikurangi

   Merupakan bentuk risiko yang dampak kerugiannya dapat dikurangi dengan cara memperkecil kemungkinan kejadiannya atau konsekuensi yang ditimbulkannya.

4. Risiko yang dipindahkan

   Merupakan bentuk risiko yang dapat dipindahkan kepada pihak lain sebagian atau keseluruhan.

2.2.6 Failure Mode and Effect Analysis (FMEA)

Failure Mode and Effect Analysis (FMEA) adalah teknik yang digunakan mendefinisikan, mengidentifikasi, dan menghilangkan masalah, kesalahan, dan lain-lain yang diketahui maupun yang berpotensi dari sistem, desain, proses maupun layanan sebelum sampai pada pelanggan [4].

FMEA merupakan metodologi sistematis yang dimaksudkan untuk melakukan kegiatan berikut:

1. Mengidentifikasi dan mengenali potensi kegagalan termasuk penyebab dan efeknya.
2. Mengevaluasi dan memprioritaskan kegagalan yang teridentifikasi.
3. Mengidentifikasi dan menyarankan tindakan yang dapat menghilangkan atau mengurangi kemungkinan terjadinya potensi kegagalan.

Mengidentifikasi kegagalan yang diketahui dan yang berpotensi adalah tugas penting dalam FMEA. Dengan menggunakan data dan pengetahuan tentang proses atau produk, setiap potensi kegagalan dan efeknya dinilai masing-masing dalam tiga faktor berikut:

1. Keparahan (*Severity*): konsekuensi kegagalan saat terjadi.
2. Kejadian (*Occurence*): probabilitas atau frekuensi kegagalan yang terjadi.
3. Deteksi (*Detection*): probabilitas kegagalan terdeteksi sebelum dampaknya terwujud.

2.2.6.1 Proses Failure Mode and Effect Analysis

Proses *Failure Mode and Effect Analysis* (FMEA) yang khas adalah metodologi proaktif yang mengikuti langkah-langkah tipikal berikut ini:

1. Pilihlah proses yang berisiko tinggi.
2. Tinjau prosesnya: langkah ini biasanya melibatkan tim yang dipilih dengan cermat yang mencakup orang-orang dengan berbagai tanggung jawab

pekerjaan dan tingkat pengalaman. Tujuan tim FMEA adalah membawa berbagai perspektif dan pengalaman ke proyek.

3. Bertukar pikiran mengenai potensi kegagalan.
4. Identifikasi akar penyebab kegagalan.
5. Sebutkan efek potensial dari setiap kegagalan.
6. Tetapkan tingkat keparahan (*severity*), kejadian (*occurence*), dan peringkat deteksi (*detection*) untuk setiap efek.
7. Hitung *Risk Priority Number* (RPN) untuk setiap efek.
8. Prioritaskan kegagalan untuk melakukan tindakan dengan menggunakan RPN.
9. Lakukan tindakan untuk menghilangkan atau mengurangi mode kegagalan berisiko tinggi.
10. Menghitung kembali RPN setelah kegagalan dikurangi atau dihilangkan sebagai alat untuk memantau produk atau proses yang didesain ulang.

2.2.6.2 Menentukan Nilai dalam Proses FMEA

Menetapkan tingkat keparahan, kejadian, dan deteksi biasanya dilakukan pada skala 1-10 di mana dijelaskan pada Tabel 2.1, Tabel 2.2, dan Tabel 2.3.

Tabel 2.1 Tingkat Keparahan

Efek	Kriteria: Tingkat Keparahan Efek	Peringkat
Berbahaya (Tanpa Peringatan)	Tingkat keparahan yang sangat tinggi bila mode kegagalan potensial mempengaruhi operasi yang aman dan / atau melibatkan ketidakpatuhan terhadap peraturan tanpa peringatan	10
Berbahaya (Dengan Peringatan)	Tingkat keparahan yang sangat tinggi bila mode kegagalan potensial mempengaruhi operasi yang aman dan / atau melibatkan ketidakpatuhan terhadap peraturan yang memperingatkan	9
Sangat Tinggi	Produk / item tidak bisa dioperasi, dengan kehilangan fungsi primer	8
Tinggi	Produk / barang bisa dioperasikan, namun pada tingkat kinerja yang rendah. Pelanggan tidak puas	7
Sedang	Produk / barang dapat dioperasikan, namun dapat menyebabkan pengerjaan ulang / perbaikan dan / atau kerusakan pada peralatan	6

Efek	Kriteria: Tingkat Keparahan Efek	Peringkat
Rendah	Produk / item dapat dioperasikan, namun dapat menyebabkan sedikit ketidaknyamanan pada operasi terkait	5
Sangat Rendah	Produk / item dapat dioperasikan, namun memiliki beberapa cacat (estetika dan sebaliknya) yang terlihat pada kebanyakan pelanggan	4
Kecil	Produk / item dapat dioperasikan, namun mungkin memiliki beberapa cacat yang terlihat oleh pelanggan yang diskriminatif	3
Sangat Kecil	Produk / barang bisa dioperasikan, namun tidak sesuai dengan kebijakan perusahaan	2
Tidak Ada	Tidak ada efek	1

Tabel 2.2 Kemungkinan Tingkat Kejadian

Kemungkinan Kegagalan	Kemungkinan Tingkat Kejadian	Peringkat
Sangat Tinggi: kegagalan hampir tak terhindarkan	≥ 1 dalam 2 kejadian	10
	1 dalam 3 kejadian	9
Tinggi: kegagalan berulang	1 dalam 8 kejadian	8
	1 dalam 20 kejadian	7
Sedang: kegagalan sesekali	1 dalam 80 kejadian	6
	1 dalam 400 kejadian	5
	1 dalam 2000 kejadian	4
Rendah: kegagalan relatif sedikit	1 dalam 15000 kejadian	3
	1 dalam 150000 kejadian	2
Sangat Rendah: kegagalan sepertinya tidak mungkin terjadi	≤ 1 dari 1500000 kejadian	1

Tabel 2.3 Tingkat Deteksi

Deteksi	Kriteria: Kemungkinan Pendeteksian Oleh Kontrol Desain	Peringkat
Mutlak Tidak Pasti	Kontrol desain tidak akan dan / atau tidak dapat mendeteksi sebab / mekanisme potensial dan mode kegagalan berikutnya; atau tidak ada kontrol desain	10
Sangat Jauh	Kemungkinan sangat jauh kontrol desain akan mendeteksi sebab / mekanisme potensial dan mode kegagalan berikutnya	9
Jauh	Kemungkinan jauh kontrol desain akan mendeteksi sebab / mekanisme potensial dan mode kegagalan berikutnya	8
Sangat Rendah	Kemungkinan sangat rendah kontrol desain akan mendeteksi sebab / mekanisme potensial dan mode kegagalan berikutnya	7

Deteksi	Kriteria: Kemungkinan Pendeteksian Oleh Kontrol Desain	Peringkat
Rendah	Kemungkinan rendah kontrol desain akan mendeteksi sebab / mekanisme potensial dan mode kegagalan berikutnya	6
Sedang	Kemungkinan sedang kontrol desain akan mendeteksi sebab / mekanisme potensial dan mode kegagalan berikutnya	5
Cukup Tinggi	Kemungkinan cukup tinggi kontrol desain akan mendeteksi sebab / mekanisme potensial dan mode kegagalan berikutnya	4
Tinggi	Kemungkinan tinggi kontrol desain akan mendeteksi sebab / mekanisme potensial dan mode kegagalan berikutnya	3
Sangat Tinggi	Kemungkinan sangat tinggi kontrol desain akan mendeteksi sebab / mekanisme potensial dan mode kegagalan berikutnya	2
Hampir Pasti	Kontrol desain hampir pasti akan mendeteksi sebab / mekanisme potensial dan mode kegagalan berikutnya	1

Setiap kegagalan, efek, dan penyebab didokumentasikan dengan menggunakan tabel seperti terlihat pada **Error! Reference source not found.** k emudian diberi penilaian untuk tingkat keparahan, kejadian, dan deteksi. Untuk mencerminkan prioritas dari kegagalan yang terdeteksi dicari nilai *Risk Priority Number* (RPN). RPN dihitung dengan mengalikan nilai keparahan (*Severity*), nilai kejadian (*Occurence*), dan nilai deteksi (*Detection*) seperti pada Persamaan 2.1.

$$RPN = Severity \times Occurrence \times Detection \quad (2.1)$$

Untuk menetapkan kategori risiko dicari nilai kritis [5]. Nilai kritis dihitung dengan membagi total nilai RPN dengan jumlah daftar risiko seperti pada Persamaan 2.2. Kategori risiko tinggi yaitu risiko yang memiliki nilai RPN lebih besar atau sama dengan nilai kritis (nilai RPN ≥ nilai kritis).

$$nilai\ kritis = \frac{total\ RPN}{jumlah\ daftar\ risiko} \quad (2.2)$$

Tabel 2.4 Dokumentasi FMEA

Kegagalan Potensial	Efek Potensial dari Kegagalan	S	Penyebab Potensial dari Kegagalan	O	Sistem Deteksi Sekarang	D	RPN	Hasil				
								Aksi yang direkomendasikan	S	O	D	RPN

2.2.6.3 Contoh Penggunaan Metode FMEA

Sebagai contoh sederhana pada kasus pengeboran lubang dilakukan analisis risiko kualitatif menggunakan metode FMEA dengan membuat dokumentasi sesuai **Error! Reference source not found.**. Setiap kegagalan d ilakukan analisis terhadap efek dan penyebabnya, selanjutnya diberi penilaian terhadap tingkat keparahan (*severity*), tingkat kejadian (*occurence*), dan tingkat deteksi (*detection*). Kemudian dihitung nilai RPN berdasarkan Persamaan 2.1. Setelah itu dapat diketahui total nilai RPN dan jumlah daftar risiko sehingga dapat diketahui nilai kritis.

$$nilai\ kritis = \frac{63 + 45 + 255}{3}$$
$$= \frac{363}{3}$$
$$= 121$$

Dari nilai kritis dapat diketahui risiko yang tergolong sebagai risiko tinggi, yaitu risiko yang memiliki nilai RPN lebih besar atau sama dengan 121. Untuk risiko yang tergolong risiko tinggi dilakukan tindakan untuk menghilangkan atau mengurangi mode kegagalan. Selanjutnya nilai RPN dihitung kembali setelah kegagalan dikurangi atau dihilangkan sebagai alat untuk memantau proses yang didesain ulang. Dokumentasi FMEA untuk kasus ini dapat dilihat pada **Error! R eference source not found.**.

Tabel 2.5 Dokumentasi FMEA untuk Contoh Kasus Pengeboran Lubang

Kegagalan Potensial	Efek Potensial dari Kegagalan	S	Penyebab Potensial dari Kegagalan	O	Sistem Deteksi Sekarang	D	RPN	Hasil				
								Aksi yang direkomendasikan	S	O	D	RPN
Lubang terlalu dalam	Menembus bagian bawah pelat	7	Penyiapan mesin yang tidak sesuai	3	Pelatihan dan instruksi pada operator	3	63					
Lubang kurang dalam	Bentuk rangkaian tidak selesai	5	Penyiapan mesin yang tidak sesuai	3	Pelatihan dan instruksi pada operator	3	45					
			Bor rusak	5	Tidak ada	9	255	Pasang alat detektor	5	5	1	25

2.2.7 Hypertext Markup Language (HTML)

Hypertext Markup Language (HTML) merupakan salah satu format yang digunakan dalam pembuatan dokumen dan aplikasi yang berjalan di halaman web. Dikatakan *markup language* karena HTML berfungsi untuk memformat *file* dokumen teks biasa untuk bisa ditampilkan pada web browser dengan bantuan tanda-tanda yang sudah ditentukan. Hal tersebut dapat dilakukan dengan menambahkan elemen atau yang sering disebut sebagai *tag* [6].

2.2.8 PHP: Hypertext Preprocessor (PHP)

PHP: *Hypertext Preprocessor* (PHP) adalah bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman *web* yang dinamis. Maksud dari *server-side scripting* adalah sintaks darn perintah-perintah yang diberikan akan sepenuhnya dijalankan di server tetapi disertakan pada dokumen HTML. Pembuatan *web* ini merupakan kombinasi antara PHP sendiri sebagai bahasa permrograman dan HTML sebagai pembangun halaman web. Ketika seorang pengguna internet akan membuka suatu situs yang menggunakan fasilitas *server-side scripting* PHP, maka terlebih dahulu server yang bersangkutan akan memproses semua perintah PHP di server lalu mengirimkan hasilnya dalam format HTML ke web browser pengguna internet tadi. Dengan demikian seorang pengguna internet tidak dapat melihat kode program yang ditulis dalam PHP sehingga keamanan dari halaman web menjadi lebih terjamin [6].

2.2.9 MySQL

MySQL adalah *multiuser database* yang menggunakan bahasa *Structured Query Language* (SQL). MySQL dalam operasi *client-server* melibatkan *server daemon* MySQL di sisi *server* dan berbagai macam program serta *library* yang berjalan di sisi *client*. MySQL mampu menangani data yang cukup besar. Perusahaan yang mengembangkan MySQL yaitu TcX, mengaku mampu menyimpan data lebih dari 40 *database*, 10.000 tabel, dan sekitar 7 juta baris, totalnya kurang lebih 100 Gigabyte data.

SQL adalah bahasa standar yang digunakan untuk mengembangkan *database server*. Bahasa ini pada awalnya dikembangkan oleh IBM, namun telah diadopsi dan digunakan sebagai standar industri. Dengan menggunakan SQL, proses akses *database* menjadi lebih *user-friendly* dibandingkan dengan menggunakan dBASE atau Clipper yang masih menggunakan perintah-perintah pemrograman [6].

2.2.10 JavaScript

JavaScript merupakan bahasa *script* populer yang dipakai untuk menciptakan halaman *web* yang dapat berinteraksi dengan pengguna dan dapat merespons *event* yang terjadi pada halaman. *JavaScript* merupakan perekat yang menyatukan halaman-halaman *web*.

JavaScript awalnya dikenal sebagai *LiveScript*, dikembangkan oleh Brendan Eich di Netscape pada tahun 1995 yang menjadi bagian terintegrasi di dalam Netscape Navigator 2.0. *JavaScript* merupakan bahasa *script* yang menghidupkan halaman-halaman HTML. *JavaScript* dapat dijalankan pada hampir semua *platform*. *JavaScript* merupakan bahasa *client-side* yang didesain pada *browser* komputer, bukan pada server. Dalam sintaksis, *JavaScript* mirip dengan C, Perl, dan Jawa.

Karena *JavaScript* terkait dengan *browser*, ia sangat terintegrasi dengan HTML. Ketika *browser* memuat sebuah halaman, *server* akan mengirim konten utuh dari dokumen, termasuk HTML dan perintah-perintah *JavaScript*. Konten HTML kemudian dibaca dan diinterpretasi baris demi baris sampai *tak* pembuka *JavaScript* dibaca, pada saat itu *intepreter JavaScript* mengambil alih. Ketika *tak* penutup *JavaScript* diraih, pemrosesan HTML berlanjut [7].

2.2.11 Cascading Style Sheet (CSS)

Cascading Style Sheet (CSS) adalah suatu cara untuk membuat format atau *layout* halaman web menjadi lebih menarik dan mudah dikelola. CSS muncul sebagai jawaban atas sulitnya mengatur *layout* tampilan dokumen web yang dibuat dengan HTML murni meskipun telah menggunakan berbagai kombinasi format [8].

2.2.12 CodeIgniter (CI)

CodeIgniter (CI) adalah *framework* pengembangan aplikasi (*Application Development Framework*) dengan menggunakan PHP, suatu kerangka untuk bekerja atau membuat program dengan menggunakan PHP yang lebih sistematis. Pemrogram tidak perlu membuat program dari awal (*from scratch*), karena CI menyediakan sekumpulan *library* yang banyak yang diperlukan untuk menyelesaikan pekerjaan yang umum, dengan menggunakan antarmuka dan struktur logika yang sederhana untuk mengakses *library*-nya. Pemrogram dapat memfokuskan diri pada kode yang harus dibuat untuk menyelesaikan suatu pekerjaan. CI merupakan *framework* yang memiliki lisensi bebas untuk digunakan, karena menggunakan lisensi *open-sorce* Apache/BSD.. CI dikembangkan pertama kali oleh Rick Ellis. [9]

2.2.13 Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah bahasa pemodelan standar untuk perangkat lunak dan pembangunan sistem. Dalam desain sistem, pemodelan dibutuhkan untuk mengelola kompleksitas. Pemodelan membantu melihat keterkaitan satu sama lain, sehingga memungkinkan untuk dapat fokus, menangkap gambaran, membuat dokumen, dan mengkomunikasikan aspek penting dari desain sistem. Dalam penggunaan UML ada 6 keuntungan yang didapat, yaitu [10]:

1. Merupakan Bahasa Formal

   Setiap elemen dari bahasa memiliki arti yang sangat mendalam, sehingga dapat meyakinkan tidak akan ada kesalahpahaman ketika memodelkan sebuah aspek tertentu dari sistem.

2. Ringkas

   Seluruh bahasa terdiri dari notasi yang sederhana dan mudah.

3. Komprehensif

   Dapat menggambarkan semua aspek penting dari sebuah sistem.

4. Ukurannya Dapat Disesuaikan

   Ketika dibutuhkan, bahasanya cukup formal untuk menangani proyek-proyek pemodelan sistem yang besar, tetapi juga dapat menangani proyek-proyek dengan skala kecil.

5. Dibangun Dari Apa yang Dipelajari

   UML merupakan puncak dari praktik-praktik terbaik di dalam komunitas berorientasi objek selama 15 tahun terakhir.

6. Merupakan Standar

   UML dikendalikan oleh kelompok standar terbuka dengan kontribusi aktif dari kelompok vendor dan kelompok akademisi di seluruh dunia.

Bagian-bagian utama dari UML adalah *view*, *diagram*, *model element*, dan *general mechanism* [11].

1. View

   View digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. *View* bukan melihat grafik, tapi merupakan suatu abstraksi yang berisi sejumlah diagram. Beberapa jenis *view* dalam UML antara lain: *use case view*, *logical view*, *component view*, *concurrency view*, dan *deployment view*.

   a. Use Case View

      Mendeskripsikan fungsionalitas sistem yang seharusnya dilakukan sesuai yang diinginkan *external actors*. *Actor* yang berinteraksi dengan sistem dapat berupa *user* atau sistem lainnya. *View* ini digambarkan dalam *use case diagram* dan terkadang dengan *activity diagram*. *View* ini digunakan terutama untuk pelanggan, perancang (*designer*), pengembang (*developer*), dan penguji sistem (*tester*).

   b. Logical View

      Mendeskripsikan bagaimana fungsionalitas dari sistem, struktur statis (*class*, *object*, dan *relationship*) dan kolaborasi dinamis yang terjadi ketika *object* mengirim pesan ke *object* lain dalam suatu fungsi tertentu. *View* ini digambarkan dalam *class diagram* untuk struktur statis dan dalam *state*, *collaboration*, dan *activity diagram* untuk model

dinamisnya. *View* ini digunakan untuk perancang (*designer*) dan pengembang (*developer*).

c. Component View

Mendeskripsikan implementasi dan ketergantungan modul. Komponen yang merupakan tipe lainnya dari *code module* diperlihatkan dengan struktur dan ketergantungannya juga alokasi sumber daya komponen dan informasi administratif lainnya. *View* ini digambarkan dalam *component view* dan digunakan untuk pengembang (*developer*).

d. Concurrency View

Membagi sistem ke dalam proses dan prosesor. *View* ini digambarkan dalam diagram dinamis (*state*, *sequence*, *collaboration*, dan *activity diagram*) dan diagram implementasi (*component* dan *deployment diagram*) serta digunakan untuk pengembang (*developer*), pengintegrasi (*integrator*), dan penguji (*tester*).

e. Deployment Ciew

Mendeskripsikan fisik dari sistem seperti komputer dan perangkat (*nodes*) dan bagaimana hubungannya dengan lainnya. *View* ini digambarkan dalam *deployment diagram* dan digunakan untuk pengembang (*developer*), pengintegrasi (*integrator*), dan penguji (*tester*).

2. Diagram

Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah *diagram* merupakan bagian dari suatu *view* tertentu dan ketika digambarkan biasanya dialokasikan untuk *view* tertentu. Adapun jenis *diagram* antara lain: *use case diagram*, *class diagram*, *state diagram*, *sequence diagram*, collaboration *diagram*, *activity diagram*, *component diagram*, dan *deployment diagram*.

a. Use Case Diagram

Menggambarkan sejumlah *external actors* dan hubungannya ke *use case* yang diberikan oleh sistem. *Use case* adalah deskripsi fungsi yang

disediakan oleh sistem dalam bentuk teks sebagai dokumentasi dari *use case symbol* namun dapat juga dilakukan dalam *activity diagram*. *Use case* digambarkan hanya yang dilihat dari luar oleh *actor* (keadaan lingkungan sistem yang dilihat *user*) dan bukan bagaimana fungsi yang ada di dalam sistem.

b. Class Diagram

Menggambarkan struktur statis *class* di dalam sistem. *Class* merepresentasikan sesuatu yang ditangani oleh sistem. *Class* dapat berhubungan dengan yang lain melalui berbagai cara: *associated* (terhubung satu sama lain), *dependent* (satu *class* tergantung/menggunakan *class* yang lain), atau *package* (grup bersama sebagai satu unit). Sebuah sistem biasanya mempunyai beberapa *class diagram*.

c. State Diagram

Menggambarkan semua *state* (kondisi) yang dimiliki oleh suatu *object* dari suatu *class* dan keadaan yang menyebabkan *state* berubah. Kejadian dapat berupa *object* lain yang mengirim pesan. *State class* tidak digambarkan untuk semua *class*, hanya yang mempunyai sejumlah *state* yang terdefinisi dengan baik dan kondisi *class* berubah oleh *state* yang berbeda.

d. Sequence Diagram

Menggambarkan kolaborasi dinamis antara sejumlah *object*. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

e. Collaboration Diagram

Menggambarkan kolaborasi dinamis seperti *sequence diagrams*. Dalam menunjukkan pertukaran pesan, *collaboration* menggambarkan *object* dan hubungannya (mengacu ke konteks). Jika penekanannya pada waktu atau urutan gunakan *sequence diagram*, tapi jika penekanannya pada konteks gunakan *collaboration diagram*.

f. Activity Diagram

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainnya seperti *use case* atau interaksi.

g. Component Diagram

Menggambarkan struktur fisik kode dari komponen, Komponen dapat berupa *source code*, komponen biner, atau *executable component*. Sebuah komponen berisi informasi tentang *logic class* atau *class* yang diimplementasikan sehingga membuat pemetaan dari *logical view* ke *component view*.

h. Deployment Diagram

Menggambarkan arsitektur fisik dari perangkat keras dan perangkat lunak sistem, menunjukkan hubungan komputer dengan perangkat (*nodes*) satu sama lain dan jenis hubungannya. Di dalam *nodes*, *executable component* dan *object* yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh *node* tertentu dan ketergantungan komponen.