

Aplikasi *Near Miss Tag* pada Departemen SKM Production PT. XYZ

Muhammad Arsi Nata Yuda¹, Henry Kurniawan², Zuriati³

¹ mahasiswa jurusan ekonomi dan bisnis, ² pembimbing 1, ³ pembimbing 2

ABSTRAK

Near miss merupakan istilah untuk peristiwa yang nyaris terjadi kecelakaan. Departemen SKM PT. XYZ saat ini mendokumentasikan *near miss* yang terjadi di lingkungan kerja dilakukan dengan cara manual, yaitu dengan menulis di form yang telah disediakan, kemudian setelah ditulis karyawan mengumpulkan form tersebut ke supervisor yang sedang bertugas saat itu, setelah dikumpulkan supervisor akan melakukan tindakan berupa peneguran atau laporan untuk perbaikan area. Cara tersebut akan menimbulkan berbagai kendala bagi karyawan, yaitu membutuhkan waktu yang lebih lama untuk melaporkan kejadian *near miss*. *Form near miss* yang telah di tulis akan disimpan di dalam lemari arsip. Data *near miss* yang disimpan di dalam lemari arsip akan membutuhkan waktu lebih lama untuk dicari. Jika data tersebut diminta untuk diolah menjadi sebuah laporan, supervisor harus mengolahnya terlebih dahulu menggunakan aplikasi pengolah angka atau pengolah kata. Sehingga pelaporan kepada manajer akan terhambat. Penelitian ini menggunakan metode pengembangan *Rapid Application Development* (RAD), karena waktu pengerjaan pembuatan aplikasi yang singkat. Hasil penelitian ini adalah dibuatnya aplikasi untuk mendokumentasikan dan melihat *progress* penyelesaian *near miss*.

Kata Kunci: *Aplikasi Near Miss Tag*, Departemen SKM Production, *Framework Codeigniter*, RAD, *form*.

PENDAHULUAN

PT. XYZ merupakan salah satu produsen susu terbaik di Indonesia yang memproduksi berbagai susu olahan, salah satunya yaitu Susu Kental Manis (SKM). Permintaan yang tinggi serta ketatnya persaingan pasar, menuntut PT. XYZ untuk terus meningkatkan jumlah produksi susu olahan yang dihasilkan. Kegiatan produksi merupakan hal yang harus selalu diperhatikan demi menghindari kecelakaan kerja.

Kecelakaan kerja yang terus meningkat akan menimbulkan kerugian bagi perusahaan, oleh sebab itu pekerja diberikan edukasi mengenai upaya untuk mengurangi jumlah kecelakaan kerja. Menurut manajer departemen SKM Production PT. XYZ kejadian dalam kecelakaan kerja dibagi menjadi empat, yaitu *fatal*, *injury accident*, *first aid* dan *near miss*.

Near miss merupakan istilah untuk peristiwa yang nyaris terjadi kecelakaan (McKinnon, 2012). Jenis *near miss* diklasifikasikan menjadi dua, yaitu disebabkan oleh perilaku tidak aman (*unsafe action*) atau kondisi tidak aman (*unsafe condition*). Departemen SKM PT. XYZ saat ini menggunakan sebuah sistem untuk mendokumentasikan *near miss* yang terjadi di lingkungan kerja. Upaya ini dilakukan agar potensi kecelakaan yang berada di lingkungan Departemen SKM PT. XYZ dapat dideteksi dan ditindaklanjuti sedini mungkin.

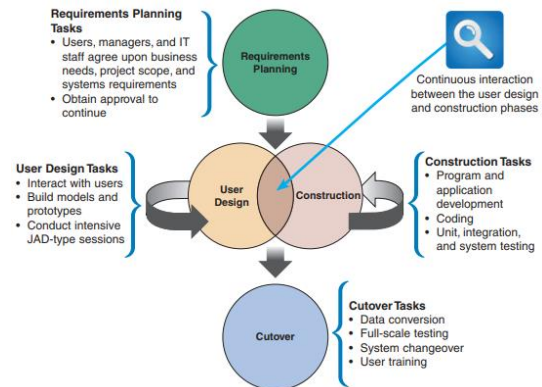
Pendokumentasian *near miss* saat ini masih dilakukan dengan cara manual, yaitu dengan menulis di *form* yang telah disediakan, kemudian setelah ditulis pekerja mengumpulkan *form* tersebut ke supervisor yang sedang bertugas saat itu, setelah dikumpulkan supervisor akan

melakukan tindakan berupa peneguran atau laporan untuk perbaikan area. Cara tersebut akan menimbulkan berbagai kendala bagi karyawan, antara lain menyulitkan karyawan untuk melaporkan kejadian karena harus mengambil *form* yang terletak di ruang supervisor. *Form near miss* yang telah di tulis akan disimpan di dalam lemari arsip. Data *near miss* yang disimpan di dalam lemari arsip akan membutuhkan waktu lebih lama untuk dicari dan diolah. Jika data tersebut diminta atau ingin diolah menjadi sebuah laporan, supervisor harus mengolahnya terlebih dahulu melalui aplikasi pengolah kata atau pengolah angka.

Permasalahan pendokumentasian dan pengolahan data *near miss* membutuhkan aplikasi yang dapat memudahkan karyawan dan supervisor dalam kegiatannya, untuk mengatasi permasalahan ini dibuatkan “Aplikasi *Near Miss* Tag pada Departemen SKM Production PT XYZ”. Aplikasi ini diharapkan dapat membantu karyawan menghemat waktu untuk melaporkan *near miss* dan mempermudah dalam membuat laporan serta pencarian data *near miss* bagi supervisor.

Metode Pelaksanaan

Metodologi pengembangan sistem yang digunakan dalam penelitian ini adalah *Rapid Application Development* (RAD). Metode RAD memiliki beberapa tahapan, yaitu *requirements planning*, *user design*, *construction* dan *cutover* (Shelly & Rosenblatt, 2017). Terlihat pada Gambar 1 dibawah ini



Gambar 1. Tahapan RAD

Hasil dan Pembahasan

Mengacu pada permasalahan mengenai dokumentasi *near miss*, maka dibuat aplikasi berdasarkan tahapan perencanaan kebutuhan dalam RAD, berupa analisis sistem yang sedang berjalan, analisis permasalahan, dan rancangan sistem baru yang akan dibuat. Tahapan RAD ini, diuraikan sebagai berikut.

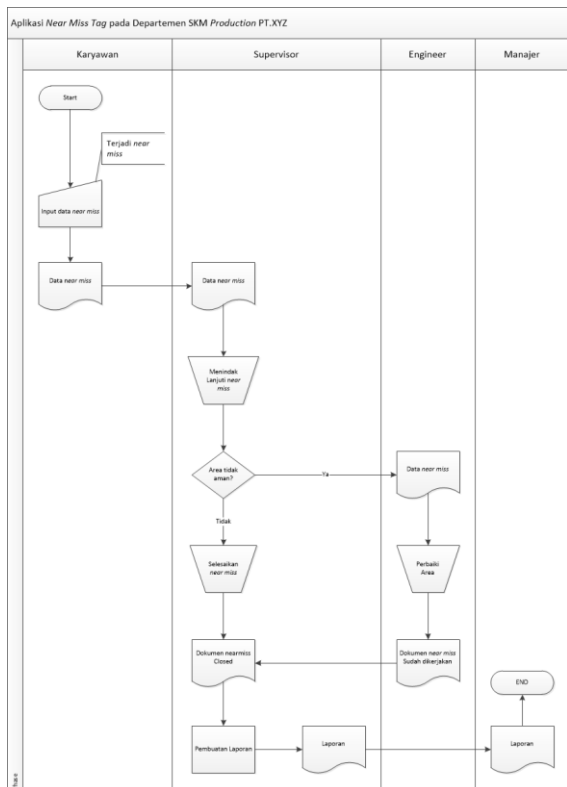
1. Requirements Planning

a. Analisis sistem yang sedang berjalan

Tahap ini menjelaskan proses pencatatan *near miss* yang berjalan di perusahaan saat ini. Analisis sistem yang sedang berjalan digambarkan dalam bentuk *Mapping Chart*.

Menurut Ismael (2017) *Mapping chart* membantu programmer menganalisis dan menyelesaikan masalah kedalam bagian yang lebih kecil dan memberikan solusi dalam melakukan analisis alternatif dalam pengoperasian.

Mapping Chart sistem yang sedang berjalan disajikan pada Gambar 2.



Gambar 2. Mapping Chart sistem lama

b. Analisis PIECES

Sebelum membangun sebuah sistem, analisa menggunakan metode PIECES (*Performance, Information Economics, Control, Efficiency, Service*) sangat penting untuk dilakukan, karena dengan analisis ini bisa ditemukan beberapa masalah utama maupun masalah yang bersifat gejala dari masalah utama. Analisis sistem menggunakan metode PIECES tersaji dalam Gambar 3.

Tahapan	Sistem yang Sedang Berjalan
<i>Performance</i>	Pencatatan data <i>near miss</i> masih dilakukan secara manual dengan menuliskan pada <i>form</i> yang hanya tersedia di lokasi tertentu.
<i>Information</i>	Informasi terkait data <i>near miss</i> tidak tersedia secara <i>real time</i> .
<i>Economics</i>	Adanya pengeluaran biaya untuk membuat <i>form</i> pencatatan <i>near miss</i> .
<i>Control</i>	Resiko hilangnya data <i>near miss</i> yang masih bersifat <i>form</i> .
<i>Efficiency</i>	Untuk mendapatkan informasi dan pembuatan laporan <i>near miss</i> diperlukan waktu yang lama.
<i>Service</i>	Jika ingin mencatat <i>near miss</i> karyawan harus pergi ke ruang supervisor untuk mengambil <i>form</i> , supervisor harus menuju ruang <i>engineer</i> untuk penyelesaian area yang tidak aman

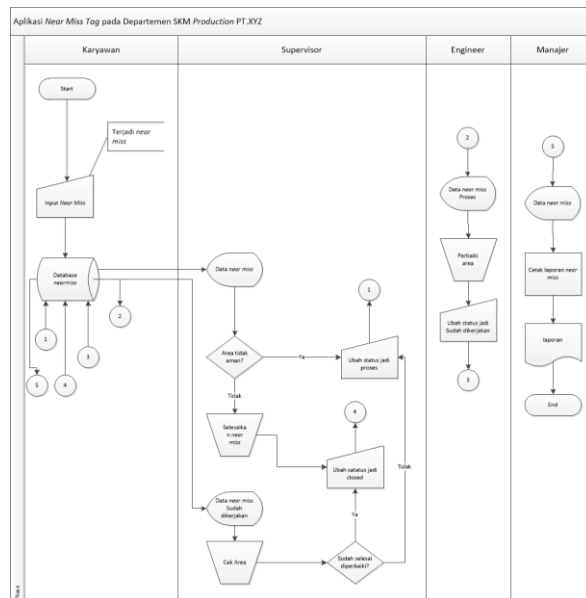
Gambar 3. Analisa PIECES

2. User Design

Penggambaran model dan *prototype* yang mewakili proses, input dan output diperoleh dari tahapan desain sistem. Tahapan ini adalah tahapan pembuatan rancangan perangkat lunak dari hasil analisis tahapan sebelumnya.

a. Mapping Chart yang diusulkan

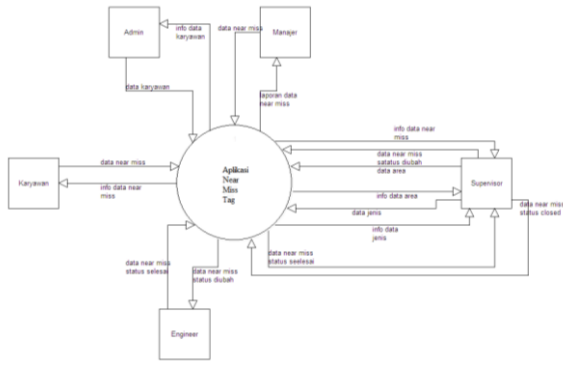
Berdasarkan analisis sistem yang sedang berjalan, ditemukan beberapa proses yang membuat sistem menjadi kurang efektif dan kurang efisien. Beberapa permasalahan tersebut ditemukan pada sistem yang sedang berjalan. Analisis sistem yang diusulkan disajikan pada Gambar 4.



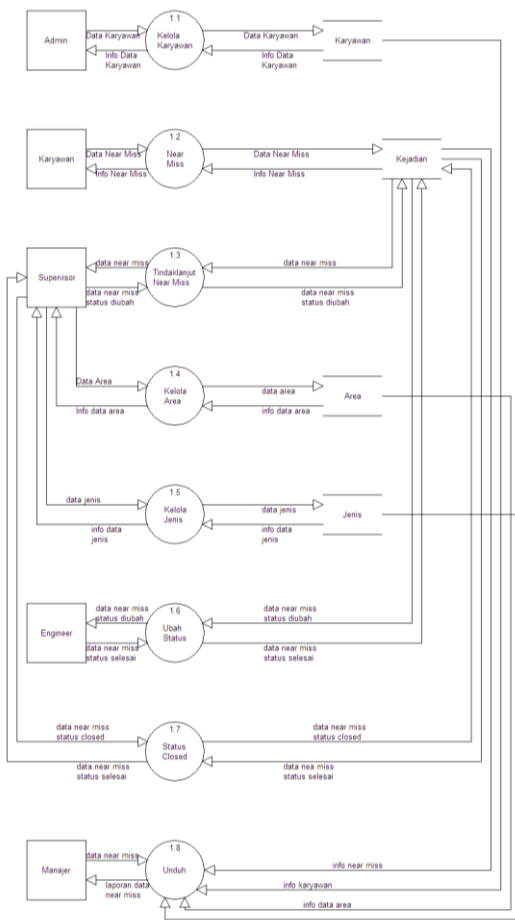
Gambar 4. Mapping Chart sistem yang diusulkan

b. Rancangan Data Flow Diagram (DFD)

DFD level 0 merupakan diagram level tertinggi yang menggambarkan hubungan sistem dengan lingkungan atau entitas luar yang terlibat dengan sistem. Entitas luar yang terlibat pada sistem ini adalah karyawan, supervisor, *engineer* dan manajer. DFD level 0 dan 1 tersaji pada Gambar 5 dan 6.



Gambar 5. DFD level 0

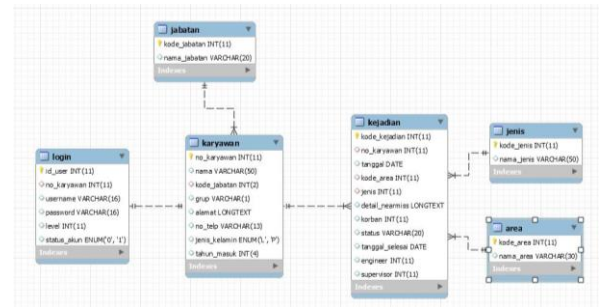


Gambar 6. DFD level 1

c. Rancangan Entity Relationship Diagram (ERD)

ERD adalah pemodelan sistem yang menunjukkan hubungan dan interaksi antara entitas sistem. ERD digunakan untuk mendesain basis data dengan tujuan menggambarkan data yang berelasi pada database (Shelly &

Rosenblatt, 2017). Rancangan ERD aplikasi *near miss tag* dapat dilihat pada Gambar 7.

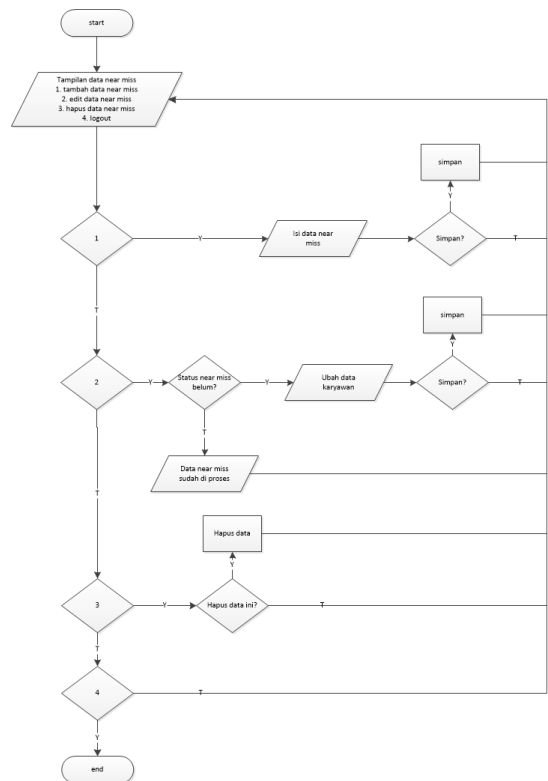


Gambar 7. Rancangan ERD aplikasi *near miss tag*

d. Rancangan flowchart

Pembuatan rancangan *flowchart* merupakan salah satu cara untuk menggambarkan algoritma aplikasi dalam bentuk simbol yang saling berhubungan. Tujuan pembuatan *flowchart* untuk mengetahui tahapan alur logika dari proses awal hingga akhir dalam suatu program.

Berikut rancangan *flowchart* aplikasi *near miss tag* dapat dilihat di bawah ini.



Gambar 8. Flowchart *near miss*

e. Desain *User Interface*

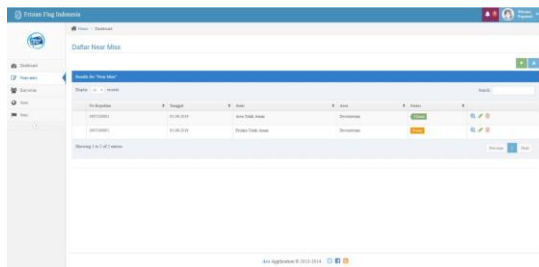
Rancangan yang akan diterapkan pada aplikasi yang akan dibuat berupa desain *interface*. Tujuan pembuatan desain *interface* agar tergambar desain aplikasi yang sesuai dengan keinginan pengguna.

3. *Construction*

Tahapan ini merupakan tahap pembuatan aplikasi yang telah di desain dengan pengkodean menggunakan bahasa pemrograman PHP, *Framework CodeIgniter*, CSS, *Javascript*, HTML, dan MySQL. Tools yang digunakan yaitu aplikasi *Sublime Text 3*, XAMPP dan *Web Browser*. Berikut ini tampilan dari aplikasi yang dibuat

a. Halaman *near miss*

Halaman *near miss* berfungsi untuk menampilkan detail dari setiap *near miss*. Informasi yang ada tentang *near miss* dengan menekan menu *near miss*.



Gambar 59. Halaman detail *near miss*

4. *Cutover*

Pada tahapan *cutover* dilakukan pengujian aplikasi secara menyeluruh, hal ini bertujuan untuk menjamin bahwa syarat dan spesifikasi rancangan sistem pada tahapan sebelumnya telah terpenuhi. Apabila terdapat kesalahan yang ditemukan, maka dilakukan perbaikan atau perubahan, sehingga aplikasi dapat digunakan oleh pengguna aplikasi.

a. Metode pengujian

Metode pengujian yang digunakan untuk menguji aplikasi ini adalah metode *black box testing* yang dilakukan dengan cara berikut :

1. Menjalankan aplikasi
2. Mengamati proses aplikasi tersebut, apakah sesuai dengan target yang tertera pada lembar pengujian yang disediakan
3. Mengisi lembar pengujian berdasarkan hasil pengamatan aplikasi

b. Hasil Pengujian

Aplikasi *near miss tag* pada departemen SKM Production PT.XYZ ini diuji oleh : Dedi Indra Setiawan (mahasiswa Politeknik Negeri Lampung), Yongki Davidson (mahasiswa Politeknik Negeri Lampung) dan Maya Norrastini (mahasiswa Politeknik Negeri Lampung).

Pengujian dengan metode *black box testing* dilakukan untuk menguji hal-hal berikut :

1. aplikasi secara fungsional,
2. akses *database*,
3. *interface* aplikasi.

Dari hasil pengujian tersebut, maka dapat diambil kesimpulan hasil pengujian sebagai berikut :

1. Aplikasi secara fungsional

Setelah dilakukan pengujian pada aplikasi *near miss tag* secara fungsional, tidak ditemukan kesalahan pada fungsi-fungsi yang ada, sehingga aplikasi berjalan dengan baik.

2. Akses *database*

Pengujian akses *database* sudah dilakukan serta diamati, dan hasilnya tidak menemukan kesalahan, data dan informasi yang

ditampilkan sesuai dengan yang diharapkan. mempermudah dalam pendokumentasian *near miss*.

3. *Interface* aplikasi

Pengujian *interface* aplikasi *near miss tag* telah dilakukan, hasilnya tidak ditemukan kesalahan dan sesuai dengan target yang tertera pada lembar pengujian.

Kesimpulan dan Saran

Kesimpulan yang dapat diambil dari pembuatan tugas akhir ini adalah menghasilkan “Aplikasi *Near Miss Tag* pada Departemen SKM *Production* PT. XYZ” yang diharapkan dapat

Refferensi

Ismael, 2017. Rancang Bangun Sistem Informasi. *Jurnal Edik Informatika*.

McKinnon, R. C., 2012. *Safety Management Near Miss Identification, Recognition, and Investgation*. s.l.:CRC Press.

Shelly, G. B. & Rosenblatt, H. J., 2017. *System Analysis and Design Ninth Edition*. Amerika: Cengage Learning.

artikel 1.docx
19 jam yang lalu

6% Risiko dari plagiarisme
LOW

Parafrase 0%
Kutipan salah 0%
Concentration ☆☆☆

↪ Bagikan

📄 Deep \$ 1.00

🔒 Monetize

🔒 View report \$ 1.50