

**RANCANG BANGUN SISTEM INFORMASI PENGELOLAAN
SURAT PERINTAH KERJA (SPK) MENGGUNAKAN
PHP 5.6.12 DAN MySQL 5.6.26 PADA
PT SUKSES CIPTA MAKMUR**

TUGAS AKHIR

Diajukan Untuk Memenuhi Syarat Program Sarjana Terapan
Program Studi Sistem Informasi Industri Otomotif Politeknik STMI Jakarta

**OLEH
AFNAN SALSABILA
1314044**



**POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA
JAKARTA
2018**

LEMBAR BIMBINGAN PENYUSUNAN TUGAS AKHIR

Nama : Afnan Salsabila

NIM : 1314044

Judul TA : Rancang Bangun Sistem Informasi Pengelolaan Surat Perintah Kerja (SPK) Menggunakan PHP 5.6.12 dan MySQL 5.6.26 pada PT Sukses Cipta Makmur

Pembimbing : Ahmad Juniar, S.Kom, MT

No	Tanggal	Keterangan	Paraf
1	23 April 2018	Bab I Pendahuluan. Perbaiki halaman 2	
2	24 April 2018	Revisi Bab I	
3	25 April 2018	Bab II Landasan Teori	
4	30 April 2018	Bab III Metodologi Penelitian	
5	30 Mei 2018	Bab IV Pengumpulan dan Pengelolaan Data. Perbaiki halaman 40, 41, 46 47, 51, 56, 61	
6	31 Mei 2018	Bab IV Pengumpulan dan Pengelolaan Data. Perbaiki halaman 65, 66, 67, 68	
7	6 Juli 2018	Revisi Bab IV	
8	23 Juli 2018	Bab V Analisis dan Pembahasan s/d <i>Use case</i> . Perbaiki halaman 73	
9	30 Juli 2018	Bab V Analisis dan Pembahasan. Perbaiki <i>Class Diagram</i>	
10	4 Agustus 2018	Bab V Analisis dan Pembahasan. <i>Deployment Diagram</i> , ERD, CDM, Kamus Data, WND, Perancangan <i>Interface</i>	
11	7 Agustus 2018	Bab VI Kesimpulan dan Saran, Daftar Pustaka	

Mengetahui,
Ketua Program Studi
Pembimbing
Sistem Informasi Industri Otomotif

Dosen

Drs. Jacob Saragih, M.M.
S.Kom, MT
NIP : 195404281986031002
197906052006041002

Ahmad Juniar,
NIP :

POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini:

Nama : Afnan Salsabila

Nim : 1314044

Program Studi : Sistem Informasi Industri Otomotif

Dengan ini menyatakan bahwa karya Tugas Akhir yang saya buat dengan judul:
“RANCANG BANGUN SISTEM INFORMASI PENGELOLAAN SURAT PERINTAH KERJA (SPK) MENGGUNAKAN PHP 5.6.12 DAN MYSQL 5.6.26 PADA PT SUKSES CIPTA MAKMUR”. Merupakan dibuat dan diselesaikan sendiri, dengan menggunakan literatur hasil kuliah, survei lapangan, dosen pembimbing dan asisten dosen pembimbing, melalui tanya jawab maupun asistensi serta buku-buku acuan yang tertera dalam referensi pada karya Tugas Akhir ini.

Jika terbukti saya tidak memenuhi apa yang telah dinyatakan di atas, maka karya Tugas Akhir saya dibatalkan.

Jakarta, 08 Agustus 2018
Yang Membuat Pernyataan,

Afnan Salsabila

ABSTRAK

PT Sukses Cipta Makmur merupakan salah satu perusahaan manufaktur yang memproduksi suku cadang untuk kebutuhan otomotif. Suku cadang untuk kebutuhan otomotif yang di produksi merupakan komponen pendukung yang dibutuhkan oleh kendaraan roda empat maupun roda dua.. Pada PT Sukses Cipta Makmur terdapat divisi PPIC yang tugasnya mengelola seluruh surat perintah kerja (SPK). SPK merupakan sebuah perintah pengerjaan yang disusun oleh departemen *Production Planning and Inventory Control* (PPIC) dan ditujukan kepada departemen produksi agar segera melakukan pengerjaan sesuai pesanan pelanggan. Proses pengelolaan surat perintah kerja masih dalam bentuk dokumen-dokumen kertas (*form*) yang ditulis tangan ke dalam rekapitulasi surat perintah kerja, sehingga mengakibatkan terjadi adanya kesalahan dalam baca tulisan. Selain itu, dalam pembuatan rekapitulasi data surat perintah kerja belum adanya sistem yang terintegrasi dengan database, sehingga terjadi keterlambatan dalam melakukan pembaruan data dan informasi selesainya perintah kerja. Sistem pengelolaan SPK sangat diperlukan dalam perusahaan untuk membantu dalam menginformasi terselesainya perintah kerja. Pengembangan sistem informasi pengelolaan SPK ini menggunakan metode *evolusionary prototype*. Pemodelan sistem menggunakan *unified modeling language* (UML), pemodelan data menggunakan *entity relationship diagram* (ERD) dan perancangan sistem menggunakan *windows navigation diagram* (WND) dan perancangan antarmuka. Pembuatan aplikasi sistem informasi pengelolaan surat perintah kerja (SPK) ini menggunakan PHP 5.6.12 dan MySQL 5.6.26. Sistem informasi pengelolaan surat perintah kerja (SPK) diharapkan dapat membantu proses pengelolaan data surat perintah kerja menjadi terkomputerisasi agar dapat berlangsung lebih akurat dalam mengenai data perintah kerja. Dan menyediakan rekapitulasi secara otomatis mengenai informasi tentang perintah kerja yang telah terselesaikan. Untuk penerapan sistem baru, disarankan untuk melakukan sosialisasi kepada bagian terkait dan pemeliharaan aplikasi agar dapat berjalan dengan baik.

Kata Kunci : pengelolaan, surat perintah kerja (SPK), *evolusionary prototype*, PHP 5.6.12, MySQL 5.6.26, UML, ERD, WND.

KATA PENGANTAR

Puji syukur dipanjatkan kepada Allah SWT, atas rahmat dan karunia-Nya sehingga penulis dapat menyusun dan menyelesaikan Tugas Akhir dengan judul **“RANCANG BANGUN SISTEM INFORMASI PENGELOLAAN SURAT PERINTAH KERJA (SPK) MENGGUNAKAN PHP 5.6.12 DAN MYSQL 5.6.26 PADA PT SUKSES CIPTA MAKMUR”**.

Penyusunan Tugas akhir ini bertujuan untuk memenuhi sebagian syarat penyelesaian Program Sarjana Terapan pada Program Studi Sistem Informasi Industri Otomotif pada Politeknik STMI Jakarta.

Terselesainya penulisan Tugas Akhir ini tidak akan terwujud tanpa adanya bantuan, bimbingan, dukungan, doa dan kasih sayang dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada :

1. Allah SWT atas rahmat dan karunia-Nya serta segala sesuatu kelancaran yang diberikan oleh-Nya dapat terselesaikan Tugas Akhir ini.
2. Kedua orang tua dan kedua adik yang selalu memberikan doa, dukungan dan kasih sayang untuk menyelesaikan Tugas Akhir ini.
3. Bapak Dr. Mustofa, S.T., M.T., selaku Direktur Politeknik STMI Jakarta.
4. Bapak Jacob Saragih, M.M., selaku Ketua Program Studi Sistem Informasi Industri Otomotif.
5. Bapak Ahmad Juniar, S.Kom, MT selaku dosen pembimbing yang telah memberikan arahan dan masukan dalam penyusunan Tugas Akhir ini.
6. Ibu Anisah dan Ibu Ambar selaku Kepala dan Wakil Kepala Divisi PPIC PT Sukses Cipta Makmur.
7. Bapak Sagus selaku pembimbing yang telah membantu mengarahkan dan membimbing selama Kerja Lapangan.
8. Ibu Ambar, Bapak Rahmat, Bapak Hermawan, Bapak Jarot, Bapak Jejen dan seluruh pegawai di PT Sukses Cipta Makmur yang telah membantu memberi arahan serta informasi untuk pengumpulan data.

9. Seluruh dosen Politeknik STMI Jakarta yang telah memberikan ilmunya guna menambah pengetahuan dan pengalaman yang sangat bermanfaat.
10. Teman-teman mahasiswa/i Politeknik STMI Jakarta Program Studi Sistem Informasi terutama kelas SA01 2014 atas kebersamaan dan motivasinya selama ini.
11. Kepada Fadhla Fanini dan Fahri Malik yang telah membantu melengkapi pembuatan tugas akhir
12. Kepada Rizkiana Kunastiti tempat konsultasi selama penyusunan tugas akhir.

Semoga Allah SWT membalas kebaikan dan ketulusan semua pihak yang telah membantu menyelesaikan laporan Tugas Akhir ini dengan melimpahkan rahmat dan karunia-Nya. Semoga penulisan ini dapat bermanfaat untuk pembaca. Terima kasih.

Jakarta, 08 Agustus 2018

Penulis,
Afnan Salsabila

BAB II

LANDASAN TEORI

2.1 Pengertian Rancang Bangun

Rancang bangun adalah tahap dari setelah analisis dari siklus pengembangan sistem yang merupakan pendefinisian dari kebutuhan-kebutuhan fungsional, serta menggambarkan bagaimana suatu sistem dibentuk yang dapat berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi, termasuk menyangkut mengkonfigurasi dari komponen-komponen perangkat keras dan perangkat lunak dari suatu sistem (Jogiyanto, 2005).

2.2 Pengertian Sistem

Sistem secara umum didefinisikan sebagai sebuah kesatuan yang kompleks, yang tersusun atas sejumlah komponen atau elemen yang saling terhubung satu sama lain, yang memudahkan di dalam jalannya satu atau beberapa buah proses. Sistem dapat disamakan dengan metode dan susunan yang teratur dari pandangan, asas dan teori-teori yang ada di dalamnya. Dalam ranah dunia komputer, mengenai tiga aspek utama, yaitu perangkat lunak komputer (*software*), perangkat keras komputer (*hardware*) dan kemampuan otak manusia (*brainware*). Ketiga aspek saling berhubungan satu sama lain, membentuk sistem operasi, tersusun atas algoritma dan pemikiran manusia (ilmuwan) dalam bentuk (*brainware*), kernel dan kelengkapan perangkat lunak lainnya (*software*), bahkan langsung *bundle* ke dalam perangkat keras komputer (*hardware*) atau memiliki dukungan terhadap akses dan komunikasi ke dalam perangkat keras komputer melalui kernelnya. Diantara ketiga aspek tersebut, terdapat prosedur yang mengatur dan mengontrol kinerja dan hubungan ketiganya (Pratama, 2016).

Kata sistem sendiri dari bahasa Latin “*Systema*” dan bahasa Yunani “*Sustema*” adalah suatu kesatuan yang terdiri komponen atau elemen yang

dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi untuk mencapai suatu tujuan. Berikut pengertian sistem menurut beberapa ahli:

1. Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran tertentu (Hutahaean, 2014).
2. Sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen atau variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu (Sutabri, 2012).
3. Sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur atau variabel-variabel yang saling terorganisasi, saling berinteraksi, dan saling bergantung sama lain (Al Fatta, 2007)

2.2.1 Karakteristik Sistem

Supaya sistem itu dikatakan sistem yang baik, harus memiliki karakteristik yaitu (Hutahaean, 2014):

1. **Komponen**
Suatu sistem terdiri dari sejumlah komponen-komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen sistem terdiri dari komponen yang berupa subsistem atau bagian-bagian dari sistem.
2. **Batasan Sistem (*boundary*)**
Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.
3. **Lingkungan Luar Sistem (*environment*)**
Lingkungan luar sistem adalah diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan dapat bersifat menguntungkan yang harus tetap dijaga dan yang merugikan harus dikendalikan sehingga tidak akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung Sistem (*interface*)

Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir antar subsistem. Keluaran (*output*) dari subsistem akan menjadi masukan (*input*) untuk subsistem lainnya melalui penghubung.

5. Masukan Sistem (*input*)

Masukan adalah energi yang dimasukkan kedalam sistem yang dapat berupa perawatan (*maintenance input*), dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan agar sistem dapat beroperasi. *Signal input* adalah energi yang untuk mendapatkan suatu keluaran.

6. Keluaran Sistem (*output*)

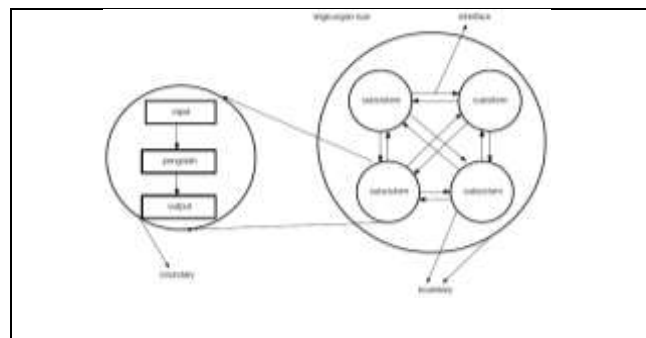
Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Contoh, komputer menghasilkan panas yang merupakan sisa pembuangan, sedangkan informasi adalah keluaran yang dibutuhkan.

7. Pengolah Sistem

Suatu sistem menjadi bagian pengolah yang akan merubah masukan menjadi keluaran. Sistem produksi akan mengolah bahan baku menjadi bahan jadi, sistem akuntansi akan mengolah data menjadi laporan keuangan.

8. Sasaran Sistem

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Sasaran dari sistem sangat menentukan *input* yang dibutuhkan sistem dan keluaran yang akan dihasilkan.



Gambar II.1 Karakteristik Sistem

Sumber: Hutahaean (2014)

2.2.2 Klasifikasi Sistem

Menurut Sutabri (2012) sistem dapat diklasifikasikan dari beberapa sudut pandang yaitu:

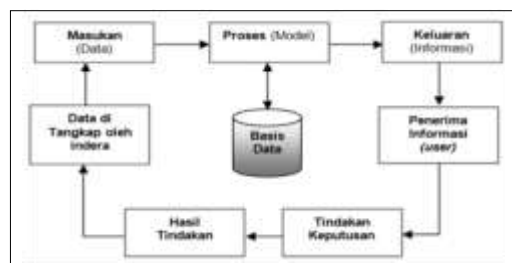
1. Sistem abstrak (*abstract system*) dan sistem fisik (*physical system*).
Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, sedangkan sistem fisik merupakan sistem yang ada secara fisik.
2. Sistem alamiah (*natural system*) dan sistem buatan manusia (*human made system*).
Sistem alamiah adalah sistem yang terjadi melalui proses alam tidak dibuat oleh manusia, sedangkan sistem buatan manusia merupakan sistem yang melibatkan interaksi manusia dengan mesin yang disebut *human made system*.
3. Sistem determinasi (*deterministic system*) dan sistem probabilistik (*probabilistic system*).
Sistem yang beroperasi dengan tingkah laku yang dapat diprediksi disebut dengan sistem determinasi, sedangkan sistem probabilistik adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilistik.
4. Sistem terbuka (*open system*) dan sistem tertutup (*closed system*).
Sistem terbuka merupakan sistem yang berhubungan dengan lingkungan luarnya, sedangkan sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh oleh lingkungan luarnya.

2.3 Pengertian Informasi

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya. Sumber informasi adalah data. Data kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian yang terjadi pada saat tertentu (Hutahaean, 2014).

Sementara menurut Sutabri (2012) informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan. Sumber dari informasi adalah data. Data merupakan kenyataan yang menggambarkan suatu kejadian serta merupakan suatu kesatuan yang nyata dan merupakan bentuk yang masih mentah sehingga perlu diolah lebih lanjut melalui suatu model sehingga menghasilkan informasi.

Menurut Hutahaean (2014), data yang diolah untuk menghasilkan informasi menggunakan model proses tertentu. Misalkan suhu dalam fahrenheit diubah ke celcius. Dalam hal ini digunakan model matematik berupa rumus konversi dari fahrenheit menjadi satuan celcius. Data yang diolah melalui suatu model menjadi informasi, kemudian penerima menerima informasi tersebut, yang berarti menghasilkan keputusan dan melakukan tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai input, diproses kembali lewat suatu model dan seterusnya yang disebut dengan siklus informasi (information cycle). Siklus ini juga disebut dengan siklus pengolahan data (data processing cycles).



Gambar II.2 Siklus Informasi

Sumber: Hutahaean (2014)

2.4 Pengertian Sistem Informasi

Sistem Informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan (Hutahaean, 2014).

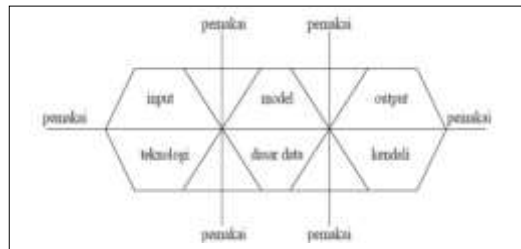
Menurut Sutabri (2012), sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi organisasi yang bersifat manajerial dalam kegiatan

strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan.

Sistem Informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building block*) (Hutahaean, 2014) yaitu:

1. Blok Masukan (*input block*)
Input mewakili data yang masuk ke dalam sistem informasi. *Input* disini termasuk metode-metode dan media yang digunakan untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen dasar.
2. Blok Model (*model block*)
Blok ini terdiri dari kombinasi prosedur, logika dan metode matematik yang akan memanipulasi data input dan data yang tersimpan dibasis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang sudah ditentukan.
3. Blok Keluaran (*output block*)
Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pengguna sistem.
4. Blok Teknologi (*technology block*)
Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian diri secara keseluruhan. Teknologi terdiri dari unsur utama: teknisi (*brainware*), perangkat lunak (*software*), dan perangkat keras (*hardware*).
5. Blok Basis Data (*data base block*)
Merupakan kumpulan dari data yang saling berhubungan satu dengan lainnya, tersimpan diperangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.
6. Blok Kendali (*control block*)
Banyak faktor yang dapat merusak sistem informasi, misalnya bencana alam, api, temperatur tinggi, air, debu, kecurangan-kecurangan, kejanggalan sistem itu sendiri, kesalahan-kesalahan, sabotase dan lain

sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah atau bila terlanjur terjadi kesalahan dapat langsung diatasi.



Gambar II.3 Blok Sistem Informasi

Sumber: Hutahaean (2015)

2.5 Pengertian Pengelolaan

Pengertian pengelolaan menurut para ahli (Saifuddin, 2014):

1. Menurut Andrew F. Sikul mengemukakan bahwa pengelolaan pada umumnya dikaitkan dengan aktivitas-aktivitas perencanaan, pengorganisasian, pengendalian, penempatan, pengarahan, pemotivasian, komunikasi dan pengambilan keputusan yang dilakukan oleh setiap organisasi dengan tujuan untuk mengkoordinasikan berbagai sumber daya yang dimiliki oleh perusahaan sehingga akan dihasilkan suatu produk atau jasa secara efisien.
2. Pengelolaan dalam bahasa Inggris berasal dari kata *management*, menurut Arifin Abdurrachman dan Purwanto (2009) mengemukakan bahwa *management* adalah kegiatan-kegiatan untuk mencapai sasaran-sasaran dan tujuan pokok yang telah ditentukan dengan menggunakan orang pelaksana.

2.6 Pengertian Surat Perintah Kerja (SPK)

Menurut Purwanto (2006), surat perintah kerja (SPK) adalah surat yang ditulis oleh pimpinan suatu perusahaan kepada pihak lain (seseorang atau organisasi), yang berisi perintah untuk mengerjakan suatu pekerjaan dalam kurun waktu tertentu sesuai dengan prosedur dan persyaratan yang telah disepakati.

Menurut Fauzi (2007), surat perintah kerja adalah surat yang isinya memberi hak serta perintah untuk melaksanakan suatu pekerjaan, sesuai dengan syarat dan ketentuan yang dijelaskan dalam perintah kerja tersebut.

Surat perintah kerja (SPK) merupakan sebuah perintah pengerjaan yang disusun oleh departemen *Production Planning and Inventory Control* (PPIC) dan ditujukan kepada departemen produksi agar segera melakukan pengerjaan sesuai pesanan pelanggan.

2.7 Pengertian Persediaan

Menurut Herjanto (2008), persediaan adalah bahan atau barang yang disimpan yang akan digunakan untuk memenuhi tujuan tertentu, misalnya untuk digunakan dalam proses produksi atau perakitan, untuk dijual kembali, atau untuk suku cadang dari suatu peralatan atau mesin. Persediaan dapat berupa bahan mentah, bahan pembantu, barang dalam proses, barang jadi, ataupun suku cadang. Bisa dikatakan tidak ada perusahaan yang beroperasi tanpa persediaan, meskipun sebenarnya persediaan hanyalah sumber dana yang menganggur karena sebelum persediaan digunakan berarti dana yang terikat didalamnya tidak dapat digunakan untuk keperluan yang lain. Persediaan dapat dikelompokkan kedalam empat jenis, yaitu:

1. *Fluctuation Stock* merupakan persediaan yang dimaksudkan untuk menjaga terjadinya fluktuasi permintaan yang tidak diperkirakan sebelumnya, dan untuk mengatasi bila terjadi kesalahan atau penyimpangan dalam prakiraan penjualan, waktu produksi, atau pengiriman barang.
2. *Anticipation Stock*, merupakan persediaan untuk menghadapi permintaan yang dapat diramalkan, misalnya pada musim permintaan tinggi tetapi kapasitas produksi pada saat itu tidak mampu memenuhi permintaan. Persediaan ini juga dimaksudkan untuk menjaga kemungkinan sukarnya diperoleh bahan baku sehingga tidak mengakibatkan terhentinya produksi.
3. *Lot-size Inventory*, merupakan persediaan yang diadakan dalam jumlah yang besar daripada kebutuhan saat itu. Persediaan dilakukan untuk

mendapatkan keuntungan dari harga barang (berupa diskon) karena membeli dalam jumlah yang besar, atau untuk mendapatkan penghematan dari biaya pengangkutan per unit lebih rendah.

4. *Pipeline Inventory*, merupakan persediaan yang dalam proses pengiriman dari tempat asal ke tempat dimana barang itu akan digunakan. Misalnya barang yang dikirim dari pabrik menuju tempat penjualan yang dapat memakan waktu beberapa hari atau minggu.

2.8 Pengertian *Material/Bahan Baku*

Material adalah zat atau benda yang dari padanya sesuatu dapat dibuat, atau barang yang dibutuhkan untuk membuat sesuatu atau istilah yang sudah mulai disosialisasikan kepada masyarakat adalah bahan baku. Material menjadi sebuah masukan dalam produksi. Jadi material adalah bahan mentah yang belum diproses, tetapi kadang kala telah diproses sebelum digunakan untuk proses produksi lebih lanjut. Umumnya, dalam masyarakat teknologi maju, material adalah bahan konsumen yang belum selesai. Beberapa contohnya adalah besi, tembaga, aluminium, kertas dan sutra dan yang lainnya (Ashby, 2005).

Menurut Mulyadi (2015) bahan baku merupakan bahan yang membentuk bagian menyeluruh produk jadi. Bahan baku yang diolah dalam perusahaan manufaktur dapat diperoleh dari pembelian lokal, *import* atau dari pengolahan sendiri.

Adapun jenis-jenis bahan mentah menurut Gunawan Adisaputro dan Marwan Asri (2016):

1. Bahan mentah langsung (*direct material*)

Bahan mentah langsung (*direct material*) adalah semua bahan mentah yang merupakan bagian barang jadi yang dihasilkan. Biaya yang dikeluarkan untuk membeli bahan mentah langsung ini mempunyai hubungan yang erat dan sebanding dengan jumlah barang jadi yang dihasilkan. Sehingga biaya bahan mentah langsung merupakan biaya variabel bagi perusahaan.

2. Bahan mentah tidak langsung (*indirect material*)

Bahan mentah tak langsung (*indirect material*) adalah bahan mentah yang ikut berperan dalam proses produksi, tetapi tidak secara langsung tampak pada barang jadi yang dihasilkan.

2.9 System Development Life Cycle (SDLC)

Menurut Dennis (2015), *System Development Life Cycle (SDLC)* memiliki empat perangkat fase dasar yaitu *planning*, *analysis*, *design* dan *implementation*. Berbeda proyek mungkin menekankan perbedaan bagian-bagian dari SDLC atau pendekatan fase SDLC dalam cara yang berbeda, tetapi semua proyek memiliki elemen dari empat fase tersebut. Masing-masing fase tersebut tersusun dari suatu rangkaian dari langkah-langkah, yang mengandalkan pada teknik yang menghasilkan dokumen spesifik dan *file* yang menjelaskan tentang proyek.

Dalam banyak proyek, fase-fase dan langkah-langkah SDLC diproses dalam suatu alur tahapan dimulai dari awal hingga akhir. Dalam proyek lain, tim proyek memindahkan langkah-langkah tersebut dengan teratur, secara bertahap, secara iteratif atau dalam pola lainnya.

Untuk saat ini, ada dua hal penting untuk dipahami tentang SDLC, yaitu:

1. Pertama, yang harus dipahami adalah mendapatkan pengertian secara umum dari fase-fase tersebut dan langkah-langkah yang digunakan dalam proyek sistem informasi dan beberapa teknik yang menghasilkan dokumen.
2. Kedua, hal yang penting untuk dipahami bahwa SDLC adalah proses penyempurnaan secara bertahap.

Hasil yang diperoleh dalam tahap analisis yang memberikan ide umur dari sistem baru. Hasil tersebut digunakan sebagai *input* pada tahap *design*, yang kemudian disempurnakan atau diperbaiki untuk menghasilkan suatu kumpulan dari dokumen yang menjelaskan detail dari sistem yang akan dibuat. Hasil kegiatan ini, pada gilirannya, digunakan dalam tahap implementasi untuk menghasilkan sistem yang sebenarnya. Setiap fase menyempurnakan dan menguraikan hasil yang telah diperoleh dari tahap sebelumnya. Berikut di bawah ini merupakan penjabaran dari setiap fase dalam SDLC:

1. *Planning* (Perencanaan)

Tahap *planning* adalah proses dasar untuk memahami mengapa sistem informasi harus dibangun dan menentukan bagaimana tim proyek akan membangunnya. Terdapat dua langkah yaitu:

- a. Selama permulaan proyek, nilai bisnis suatu sistem terhadap organisasi diidentifikasi: apakah sistem tersebut akan menurunkan biaya atau meningkatkan pendapatan? Sebagian besar gagasan untuk sistem baru berasal dari luar Departemen IT (misalnya Departemen Pemasaran, Departemen Akuntansi) dalam bentuk permintaan sistem. Permintaan sistem menyajikan ringkasan singkat tentang kebutuhan bisnis dan ini menjelaskan bagaimana sistem yang mendukung kebutuhan akan menciptakan nilai bisnis. Departemen IT bekerja sama dengan orang atau Departemen yang menghasilkan permintaan (sponsor proyek) untuk melakukan analisis kelayakan informasi.

Permintaan sistem dan analisis kelayakan disajikan kepada pihak yang dapat memberikan persetujuan, yang menentukan apakah proyek dilaksanakan atau tidak.

- b. Setelah proyek disetujui, langkah selanjutnya adalah manajemen proyek. Selama manajemen proyek, manajer proyek membuat sebuah rencana kerja, memilih anggota proyek tersebut dan menentukan teknik-teknik untuk membantu tim proyek mengendalikan dan mengarahkan proyek pada keseluruhan tahap SDLC. Hasil dari manajemen proyek adalah rencana proyek yang menjelaskan bagaimana tim proyek akan mengembangkan sistemnya.

2. *Analysis* (Analisis)

Selama tahap ini, tim proyek menyelidiki sistem yang ada saat ini, mengidentifikasi peluang untuk perbaikan dan mengembangkan sistem baru. Tahap ini memiliki tiga langkah:

- a. Strategi analisis yang dikembangkan untuk mengarahkan tim proyek. Seperti misalnya strategi yang termasuk untuk menganalisis sistem yang telah ada (disebut *as-is-system*) beserta masalah-masalah yang ada untuk merancang sistem (disebut sebagai *to-be-system*).

- b. Langkah selanjutnya adalah pengumpulan persyaratan (misalnya, melalui wawancara atau kuesioner). Analisis dari informasi ini bersamaan dengan masukan dari sponsor proyek dan banyak orang lainnya mengarah pada pengembangan konsep untuk sistem. Konsep sistem ini kemudian digunakan sebagai dasar untuk mengembangkan seperangkat model analisis bisnis, yang menggambarkan bagaimana bisnis akan beroperasi jika sistem baru telah dikembangkan.
- c. Analisis, konsep sistem dan model digabungkan menjadi dokumen yang disebut proposal sistem, yang dipresentasikan ke sponsor proyek dan pengambil keputusan utama lainnya yang memutuskan apakah proyek harus terus dilaksanakan.

3. *Design* (Perancangan)

Tahap *design* memutuskan bagaimana sistem akan beroperasi, dalam hal perangkat keras, perangkat lunak dan infrastruktur jaringan; antarmuka pengguna, formulir dan laporan; program spesifik, *database* dan *file* yang akan dibutuhkan. Meskipun sebagian besar keputusan strategis mengenai sistem dibuat dalam pengembangan konsep sistem selama tahap analisis, langkah-langkah dalam tahap *design* menentukan secara tepat bagaimana sistem akan beroperasi. Fase *design* memiliki empat langkah:

- a. Strategi *design* yang pertama kali dikembangkan. Hal tersebut menjelaskan apakah sistem akan dikembangkan oleh *programmer* perusahaan sendiri, apakah sistem akan diberikan ke perusahaan lain (perusahaan konsultan), atau apakah perusahaan akan membeli paket perangkat lunak yang ada.
- b. Pengembangan *design* yang menggambarkan perangkat keras, perangkat lunak dan infrastruktur jaringan yang akan digunakan. Dalam kebanyakan kasus, sistem akan menambah atau mengubah infrastruktur yang sudah ada dalam organisasi. Desain antarmuka menentukan bagaimana pengguna akan bergerak melalui sistem (misalnya, metode navigasi seperti menu dan tombol di layar), formulir dan laporan yang akan digunakan sistem.

- c. Pengembangan *database* dan spesifikasi *file*. Pengembangan ini menentukan dengan tepat data apa yang akan disimpan dan dimana akan disimpan.
 - d. Tim analisis mengembangkan desain program, yang akan mendefinisikan program yang perlu ditulis dan yang akan dilakukan masing-masing program.
4. *Implementation* (Implementasi)
- Tahap akhir dalam SDLC adalah tahap *implementation*, dimana sistem benar-benar dibangun. *Implementation* adalah fase yang paling mendapat perhatian, karena kebanyakan sistem merupakan bagian terlama dan paling mahal dari proses pembangunan. Fase ini memiliki tiga langkah:
- a. Pertama merekonstruksi sistem. Sistem ini dibangun dan diuji untuk memastikan kinerjanya sebagaimana yang telah dirancang. Karena biaya *bugs* bisa sangat besar, pengujian merupakan salah satu langkah paling kritis dalam implementasi. Sebagian besar organisasi member lebih banyak waktu dan perhatian untuk menguji daripada menulis program.
 - b. Instalasi sistem. Instalasi adalah proses dimana sistem lama dimatikan dan yang baru diaktifkan. Salah satu aspek terpenting adalah pengembangan rencana pelatihan untuk mengajarkan pengguna cara menggunakan sistem baru dan membantu mengelola perubahan yang disebabkan oleh sistem yang baru.
 - c. Menetapkan rencana dukungan untuk sistem yang dilakukan oleh tim analisis. Rencana ini biasanya mencakup tinjauan pasca-pelaksanaan formal atau informal serta cara sistematis untuk mengidentifikasi perubahan besar dan kecil yang diperlukan untuk sistem tersebut.

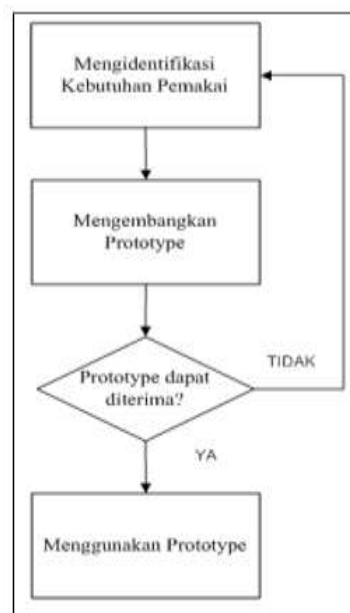
2.10 Model *Prototype*

Menurut McLeod dan Schell (2012), *prototype* adalah suatu versi sistem potensial yang disediakan bagi pengembang dan calon pengguna yang dapat memberikan gambaran bagaimana kira-kira sistem tersebut akan berfungsi bila

telah disusun dalam bentuk yang lengkap. Proses dalam memproduksi suatu *prototype* disebut *prototyping*. Tujuannya adalah menghasilkan *prototype* secepat mungkin dan memperoleh umpan balik dari pengguna yang akan memungkinkan prototipe untuk ditingkatkan sampai sistem dianggap sempurna. Adapun jenis *prototype* terdapat dua jenis *prototype* yaitu *prototype* evolusioner (*evolutionary prototype*) dan *prototype* requirement (*requirement prototype*).

2.10.1 *Prototype Evolusioner*

Prototype evolusioner adalah *prototype* yang terus menerus diperbaiki sampai semua kriteria sistem yang baru terpenuhi. Ada empat langkah yang diambil dalam mengembangkan suatu *prototype evolusioner* yaitu identifikasi kebutuhan pengguna, mengembangkan prototipe, menentukan *prototype* dapat diterima atau tidak, dan penggunaan *prototype*.



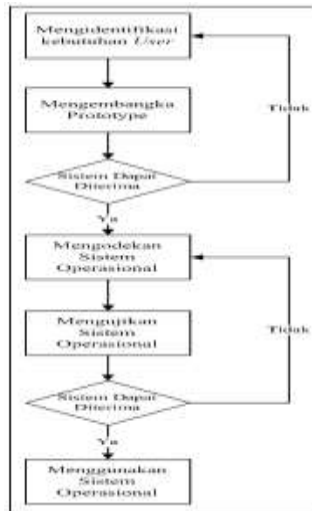
Gambar II.4 Pengembangan *Prototype Evolusioner*

(Sumber: McLeod, 2012)

2.10.2 *Prototype Requirement*

Prototype requirement adalah suatu pengembangan untuk menentukan kebutuhan fungsional dari sistem baru pada saat para pengguna tidak mampu mengungkapkan dengan tepat apa yang mereka butuhkan. Saat kebutuhan telah

ditentukan *prototype* requirement dapat mulai dikerjakan dan proyek siap untuk mengembangkan suatu sistem yang baru.



Gambar II.5 Pengembangan *Prototype Requirement*

(Sumber: McLeod, 2012)

2.11 *Flowchart*

Menurut Andika (2016) mendefinisikan *flowchart* sebagai bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. *Flowchart* merupakan cara penyajian dari suatu algoritma.

Flowchart atau diagram alir merupakan sebuah diagram dengan simbol-simbol grafis yang menyatakan aliran algoritma atau proses yang menampilkan langkah-langkah yang disimbolkan dalam bentuk kotak, beserta urutannya dengan menghubungkan masing-masing langkah tersebut menggunakan tanda panah. Diagram ini bisa memberikan solusi selangkah demi selangkah untuk menyelesaikan masalah yang ada dalam proses atau algoritma tersebut (Wikipedia, 2017).

2.11.1 Macam-macam *Flowchart*

Menurut Jogiyanto (2005), terdapat lima macam *flowchart*, yaitu sebagai berikut:

1. Bagan Alir Sistem (*Systems Flowchart*)

Merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan. Bagan ini menjelaskan urutan-urutan dari prosedur-prosedur yang ada di dalam sistem.

2. Bagan Alir Dokumen (*Document Flowchart*)

Bagan alir dokumen atau disebut juga bagan alir formulir (*form flowchart*) atau *paperwork flowchart* merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya. Bagan alir dokumen ini menggunakan simbol-simbol yang sama dengan yang digunakan di dalam bagan alir sistem.

3. Bagan Alir Skematik (*Schematic Flowchart*)

Merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu untuk menggambarkan prosedur di dalam sistem. Perbedaannya adalah, bagan alir skematik selain menggunakan simbol-simbol bagan alir sistem juga menggunakan gambar-gambar komputer dan peralatan lainnya yang digunakan. Maksud penggunaan gambar-gambar ini adalah untuk memudahkan komunikasi kepada orang yang kurang pahan dengan simbol-simbol bagan alir. Penggunaan gambar-gambar ini memudahkan untuk dipahami, tetapi sulit dan lama menggambarinya.

4. Bagan Alir Program (*Program Flowchart*)

Merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program. Bagan alir program dapat terdiri dari dua macam, yaitu Bagan Alir Logika Program (*Program Logic Flowchart*) dan Bagan Alir Program Komputer Terinci (*Detailed Computer Program Flowchart*).

5. Bagan Alir Proses (*Process Flowchart*)




Merupakan bagan alir yang banyak digunakan di teknik industri. Bagan alir ini juga berguna bagi analisis sistem untuk menggambarkan proses dalam suatu prosedur. Bagan alir proses selain dapat menunjukkan kegiatan dan simpanan yang digunakan dalam suatu prosedur, dapat juga menunjukkan jarak kegiatan yang satu dengan yang lainnya serta waktu yang diperlukan oleh suatu kegiatan.

2.11.2 Simbol-simbol *Flowchart*

Menurut Andika (2016) simbol-simbol yang dipakai dalam *flowchart* dibagi menjadi tiga kelompok, yakni:

1. *Flow Direction Symbols*. Digunakan untuk menghubungkan simbol satu dengan yang lain disebut juga *connecting line*.





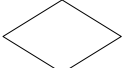
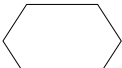
Tabel II.1 *Flow Direction Symbols*

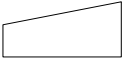
Nama	Simbol	Keterangan
<i>Arus/flow</i>		Menyatakan jalannya arus suatu proses
<i>Connector</i>		Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama
<i>Offline connector</i>		Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda

Sumber: Andika (2016)

2. *Processing Symbols*. Menunjang jenis operasi pengolahan dalam suatu proses atau prosedur *input* atau *output*.

Tabel II.2 *Processing Symbols*

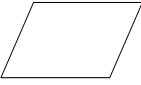



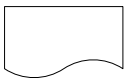

Nama	Simbol	Keterangan
<i>Process</i>		Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer
<i>Manual process</i>		Menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer
<i>Terminal</i>		Menyatakan permulaan atau akhir suatu program
<i>Offline storage</i>		Menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu
<i>Decision</i>		Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban ya atau tidak
<i>Predefined process</i>		Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal

<i>Manual input</i>		Memasukkan data secara manual dengan menggunakan <i>online keyboard</i>
---------------------	---	---

Sumber: Andika (2016)

3. *Symbols*. Menunjukkan jenis peralatan yang digunakan sebagai media *input* atau *output*.

Tabel II.3 *Symbols*

Nama	Simbol	Keterangan
<i>Input</i> atau <i>output</i>		Menyatakan proses <i>input</i> atau <i>output</i> tanpa tergantung jenis peralatannya
<i>Punched card</i>		Menyatakan <i>input</i> berasal dari kartu atau <i>output</i> ditulis ke kartu
<i>magnetic tape</i>		Menyatakan <i>input</i> berasal dari pita magnetis atau <i>output</i> disimpan ke pita magnetis
<i>disk storage</i>		Menyatakan <i>input</i> berasal dari <i>disk</i> atau <i>output</i> disimpan ke <i>disk</i>
<i>document</i>		Mencetak keluaran dalam bentuk dokumen (melalui printer)
<i>display</i>		Mencetak keluaran dalam layar monitor

Sumber: Andika (2016)

2.12 *System Requirements*

System requirements atau kebutuhan sistem adalah semua aktivitas yang harus dilakukan atau didukung oleh sistem baru dan batasan-batasan yang harus dicapai sistem baru (Satzinger, 2012). *System requirements* dibagi menjadi 2 kategori, yaitu:

1. *Functional Requirement* merupakan penjelasan tentang layanan yang perlu disediakan oleh sistem, bagaimana sistem menerima dan mengolah masukan, dan bagaimana sistem mengatasi situasi-situasi tertentu. Selain itu kadang-kadang juga secara jelas menentukan apa yang tidak dikerjakan oleh sistem. *Functional Requirement* menggambarkan *system requirement* secara detail seperti input, output dan pengecualian yang berlaku.

2. *Non-Functional Requirement* secara umum berisi batasan-batasan pada pelayanan atau fungsi yang disediakan oleh sistem. Termasuk di dalamnya adalah batasan waktu, batasan proses pembangunan, standar-standar tertentu. Karena berkaitan dengan kebutuhan sistem secara keseluruhan, maka kegagalan memenuhi kebutuhan jenis ini berakibat pada sistem secara keseluruhan.

2.13 *Unified Modeling Language (UML)*

Unified Modelling Language (UML) adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek. Tujuan dari UML adalah untuk menyediakan kosa kata yang umum dari istilah-istilah berbasis objek dan teknik yang cukup banyak untuk memodelkan proyek pengembangan sistem dari analisis ke desain. Menurut Dennis et al (2010), diagram-diagram yang ada dalam UML terbagi menjadi dua bagian utama yaitu, *Struktur diagram* dan *Behavior diagram*. *Structure diagram* biasanya digunakan untuk merepresentasikan data dan hubungan statik yang ada di dalam sebuah sistem informasi. Sedangkan *Behavior diagram* menyediakan para analis dengan sebuah gambaran hubungan yang dinamis antara instansi atau objek yang merepresentasikan sistem informasi bisnis. Berikut adalah tabel dari jenis diagram yang ada dan fungsi-fungsinya:

Tabel II.4 Jenis Diagram Pada UML dan Fungsi-fungsinya

Nama Diagram	Fungsi
<i>Structure Diagrams</i>	
<i>Class Diagram</i>	Menggambarkan hubungan antara pemodelan-pemodelan <i>class</i> di dalam sistem.
<i>Object Diagram</i>	Menggambarkan hubungan antara pemodelan-pemodelan objek di dalam sistem.
<i>Package Diagram</i>	Mengelompokkan elemen UML lainnya menjadi satu untuk membentuk level konstruksi yang lebih tinggi.

<i>Deployment Diagram</i>	Menampilkan arsitektur fisik dari suatu sistem. Bisa juga digunakan untuk menunjukkan komponen <i>software</i> sedang dijadikan arsitektur fisik suatu sistem.
---------------------------	--

Tabel II.4 Jenis Diagram Pada UML dan Fungsi-fungsinya (lanjutan)

Nama Diagram	Fungsi
<i>Component Diagram</i>	Menggambarkan hubungan fisik diantara komponen-komponen <i>software</i> .
<i>Composite Diagram</i>	Menggambarkan struktur internal dari suatu <i>class</i> dan hubungan diantara bagian-bagian dari suatu <i>class</i> .
<i>Behavior Diagram</i>	
<i>Activity Diagram</i>	Menggambarkan proses bisnis masing-masing <i>class</i> .
<i>Sequence Diagram</i>	Memodelkan kebiasaan dari objek-objek dalam suatu <i>use case</i> . Fokus dalam aktifitas berdasarkan urutan waktu.
<i>Communication Diagram</i>	Memodelkan kebiasaan dari objek-objek dalam suatu <i>use case</i> . Fokus dalam komunikasi antara satu set dari kolaborasi objek dari suatu objek.
<i>Interaction Overview Diagram</i>	Menggambarkan suatu ikhtisar alur dari kontrol suatu proses.
<i>Timing Diagram</i>	Menggambarkan interaksi yang terjadi diantara suatu set dari objek-objek dan perubahan keadaan selama perjalanan waktu.
<i>Behavioral State Machine Diagram</i>	Memeriksa kebiasaan dari suatu <i>class</i> .
<i>Protocol State Machine Diagram</i>	Menggambarkan ketergantungan diantara perbedaan-perbedaan <i>interface</i> dari suatu <i>class</i> .
<i>Protocol State Machine Diagram</i>	Menggambarkan ketergantungan diantara perbedaan-perbedaan <i>interface</i> dari suatu <i>class</i> .
<i>Use Case Diagram</i>	Menangkap kebutuhan bisnis untuk sistem dan untuk menggambarkan interaksi diantara sistem dan lingkungannya.

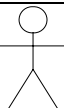
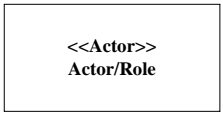
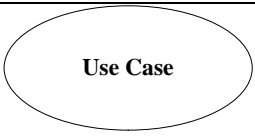
(Sumber: Dennis et al, 2015)

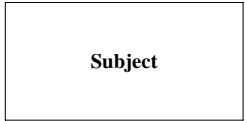
2.13.1 Use Case Diagram

Diagram *use case* dibuat untuk menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Suatu landasan pola pikir yang ditekankan dalam diagram *use case* ini adalah “apa” yang dapat diperbuat oleh sistem, dan bukan “bagaimana” sistem melakukannya.


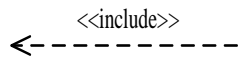
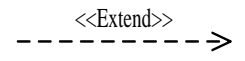

Diagram *use case* menggambarkan kebutuhan sistem dari sudut pandang *user*, yaitu proses yang dilakukan oleh sistem dalam melayani *user* yang berinteraksi dengan sistem tersebut. Diagram ini sangat tepat untuk menggambarkan hubungan yang terjadi antara sistem dengan *user*nya (Dennis et al, 2015).

Tabel II.5 Simbol-simbol *Use Case Diagram*

Simbol	Fungsi
 <p>Actor/Role</p> 	<ul style="list-style-type: none"> • Menyatakan seseorang atau sistem yang mendapatkan keuntungan dari sistem. • Digambarkan sebagai gambar <i>stick</i> /gambar orang (<i>default</i>) atau jika bukan seorang aktor manusia, digambarkan dengan suatu kotak dengan tanda <<actor>> di dalamnya (alternatif). • Dilabelkan dengan peran/<i>role</i> dari aktor. • Dapat diasosiasikan dengan aktor menggunakan asosiasi spesialisasi/ <i>superclass</i> (<i>specialization/superclass association</i>). • Ditempatkan di luar batas sistem.
	<ul style="list-style-type: none"> • Merepresentasikan bagian utama dari fungsionalitas suatu sistem. • Dapat berupa perluasan <i>use case</i> lain. • Dapat termasuk di dalam <i>use case</i> lain. • Diletakan di dalam batas sistem. • Dinamakan dengan frasa kata kerja.

	<ul style="list-style-type: none"> • Menyertakan nama subjek di dalam maupun di atas. • Merepresentasikan ruang lingkup dari subjek, sistem atau proses bisnis.
---	---

Tabel II.5 Simbol-simbol *Use Case Diagram* (lanjutan)

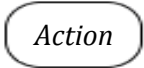
Simbol	Fungsi
 Assosiasi	<ul style="list-style-type: none"> • Menghubungkan suatu aktor dengan <i>use case</i> dengan interaksi antara keduanya.
 <<include>>	<ul style="list-style-type: none"> • Merepresentasikan fungsionalitas suatu <i>use case</i> dengan <i>use case</i> lainnya. • Disimbolkan dengan anak panah dari sebuah <i>use case</i> dasar ke <i>use case</i> yang digunakan.
 <<Extend>>	<ul style="list-style-type: none"> • Merepresentasikan perluasan (<i>extend</i>) dari <i>use case</i> lain untuk menyertakan perilaku opsional / <i>optional</i> (tidak wajib). • Disimbolkan dengan anak panah yang digambarkan dari perluasan <i>use case</i> ke <i>use case</i> dasar.
 Generalisasi	<ul style="list-style-type: none"> • Merepresentasikan <i>use case</i> khusus ke satu <i>use case</i> yang lebih umum. Disimbolkan dengan anak panah.

(Sumber: Dennis et al, 2015)


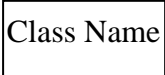

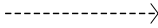



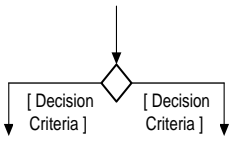
2.13.2 *Activity Diagram*

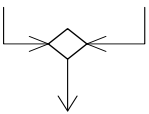
Digunakan untuk memodelkan perilaku dalam proses bisnis yang independen dari objek. Memodelkan segala sesuatu mulai dari alur kerja bisnis tingkat tinggi yang melibatkan banyak kasus penggunaan yang berbeda, hingga detail kasus penggunaan individual, sampai ke detail yang spesifik dari metode individual. Singkatnya, dapat digunakan untuk memodelkan semua jenis proses (Dennis et al, 2015). Berikut adalah simbol-simbol yang ada pada *Activity Diagram*:

Tabel II.6 Simbol-simbol *Activity Diagram*

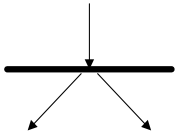
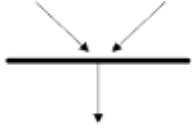
Simbol	Deskripsi
	<i>Action</i> Perilaku yang sederhana, tidak dapat diuraikan. Diberi label dengan namanya.

Tabel II.6 Simbol-simbol *Activity Diagram* (lanjutan)

Simbol	Deskripsi
	<i>Activity</i> Digunakan untuk mewakili serangkaian tindakan. Diberi label berdasarkan nama.
	<i>Object Node</i> Digunakan untuk mewakili objek yang terhubung ke satu set arus objek. Diberi label oleh nama kelasnya.
	<i>Control Flow</i> Menunjukkan urutan langkah.
	<i>Object Flow</i> Menunjukkan aliran suatu objek dari satu aktivitas (atau tindakan) ke aktivitas lain (atau tindakan).
	<i>Initial Node</i> Menggambarkan awal dari serangkaian tindakan/kegiatan.
	<i>Final-Activity Node</i> Digunakan untuk menghentikan semua aliran kontrol dan arus objek dalam suatu aktivitas.
	<i>Final-Flow Node</i> Digunakan untuk menghentikan aliran kontrol spesifik atau aliran objek.
	<i>Decision Node</i> Digunakan untuk mewakili kondisi pengujian untuk memastikan bahwa aliran kontrol atau aliran objek hanya turun satu jalur. Diberi label dengan kriteria keputusan untuk melanjutkan jalur tertentu.

	<p><i>Merge Node</i></p> <p>Digunakan untuk membawa kembali jalur keputusan yang berbeda yang dibuat menggunakan simpul keputusan.</p>
---	--

Tabel II.6 Simbol-simbol *Activity Diagram* (lanjutan)

Simbol	Deskripsi
	<p><i>Fork Node</i></p> <p>Digunakan untuk membagi perilaku menjadi seperangkat aktivitas yang paralel atau bersamaan (atau tindakan).</p>
	<p><i>Join Node</i></p> <p>Digunakan untuk membawa kembali seperangkat aktivitas yang paralel atau bersamaan dari aktivitas (atau tindakan).</p>
<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;">SwimLane</div>	<p><i>Swimlane</i></p> <p>Digunakan untuk memecah diagram aktivitas menjadi baris dan kolom untuk menetapkan aktivitas individu (atau tindakan) kepada individu atau objek yang bertanggung jawab untuk melaksanakan aktivitas (atau tindakan). Diberi label dengan nama individu atau objek yang bertanggung jawab</p>

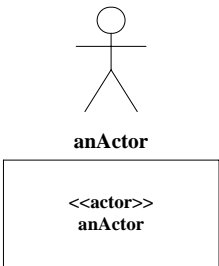
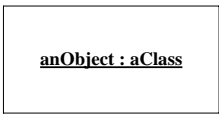


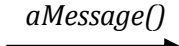
(Sumber: Dennis et al, 2015)

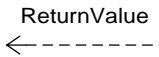
2.13.3 *Sequence Diagram*

Sequence diagrams adalah salah satu dari dua jenis diagram interaksi. Mereka mengilustrasikan objek yang berpartisipasi dalam *use case* dan pesan yang lewat di antara mereka dari waktu ke waktu untuk satu *use case*. *Sequence diagram* adalah model dinamis yang menunjukkan urutan eksplisit pesan yang dilewatkan antara objek dalam interaksi yang ditentukan. Karena *sequence diagrams* menekankan urutan waktu dari aktivitas yang terjadi di antara sekumpulan objek, mereka sangat membantu untuk memahami spesifikasi waktu nyata dan kasus penggunaan yang kompleks.

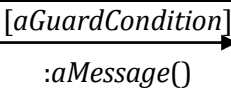

Sequence diagrams bisa menjadi diagram urutan generik yang menunjukkan semua skenario yang mungkin untuk kasus penggunaan, tetapi biasanya setiap analis mengembangkan seperangkat contoh *sequence diagrams*, masing-masing yang menggambarkan satu skenario dalam kasus penggunaan. Jika tertarik untuk memahami alur kontrol skenario berdasarkan waktu, harus menggunakan *sequence diagram* untuk menggambarkan informasi ini. Diagram digunakan sepanjang tahap analisis dan desain. Namun, diagram desain sangat spesifik implementasinya, sering termasuk objek database atau komponen antarmuka pengguna tertentu sebagai objek. (Dennis et al, 2015). Berikut simbol yang ada pada *Sequence Diagram*:

Tabel II.7 Simbol-simbol *Sequence Diagram*

Simbol	Deskripsi
	<p><i>An Actor</i></p> <p>orang atau sistem yang memperoleh manfaat dari dan berada di luar sistem. Berpartisipasi dalam suatu urutan dengan mengirim dan / atau menerima pesan. Ditempatkan di bagian atas diagram. Digambarkan baik sebagai alternatongkat (default) atau, jika aktor bukan manusia terlibat, sebagai persegi panjang dengan << aktor >> di dalamnya (alternative).</p>
	<p><i>An Object</i></p> <p>Berpartisipasi dalam suatu urutan dengan mengirim dan / atau menerima pesan. Ditempatkan di bagian atas diagram.</p>
	<p><i>A Lifeline</i></p> <p>Menunjukkan kehidupan suatu objek selama suatu urutan. Berisi X pada titik di mana kelas tidak lagi berinteraksi.</p>
	<p><i>An Execution Occurrence</i></p> <p>persegi panjang panjang yang sempit yang menempatkan garis hidup atas. Menunjukkan kapan suatu objek mengirim/menerima pesan.</p>
	<p><i>A Message</i></p>

	<p>Menyampaikan informasi dari satu objek ke objek lainnya. Panggilan operasi diberi label dengan pesan yang dikirim dan panah padat, sedangkan pengembalian diberi label dengan nilai yang dikembalikan dan ditampilkan sebagai tanda panah putus-putus.</p>
---	---

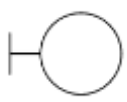

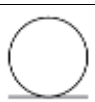
Tabel II.7 Simbol-simbol *Sequence Diagram* (lanjutan)

Simbol	Deskripsi
	<p><i>A Guard Condition</i> Merupakan tes yang harus dipenuhi untuk pesan yang akan dikirim.</p>
<p>X</p>	<p><i>For Object Destruction</i> X ditempatkan di ujung garis hidup objek untuk menunjukkan bahwa itu akan keluar dari keberadaan.</p>
	<p><i>A Frame</i> Menunjukkan urutan konteks diagram.</p>

(Sumber: Dennis, 2010)

Selain simbol-simbol diatas dalam buku *Secure Software Design*, Richardson dan Thies (2013) mendefinisikan *sequence diagram* memiliki *class* dengan fungsi yang masing-masing berbeda, berikut *class* yang terdapat dalam *class sequence diagram* menurut Richardson dan Thies:

Tabel II.8 *Class Sequence Diagram*

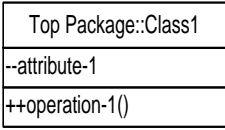

Simbol	Istilah dan Definisi
	<p><i>Boundary class:</i> <i>class</i> yang menangani interaksi antara aktor dengan sistem (<i>user interface</i>).</p>
	<p><i>Control class:</i> control class adalah coordinator untuk sistem. Tempat logika sistem diimplementasikan.</p>
	<p><i>Entity class:</i> <i>Entity class</i> secara garis besar adalah <i>class</i> penyimpanan.</p>

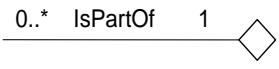
(sumber: Richardson, 2013)

2.13.4 Class Diagram


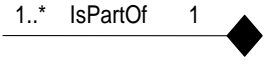
Class diagram adalah model statis yang menunjukkan kelas dan hubungan di antara kelas yang tetap konstan dalam sistem dari waktu ke waktu. *Class diagram* menggambarkan kelas, yang mencakup perilaku dan status, dengan hubungan antar kelas. Bagian berikut menyajikan elemen-elemen diagram kelas, pendekatan berbeda yang dapat digunakan menyederhanakan diagram kelas, dan diagram struktur alternatif: diagram objek (Dennis et al, 2015). Berikut simbol yang ada pada *class diagram*:

Tabel II.9 Simbol-simbol *Class Diagram*

Simbol	Deskripsi
	<p><i>A Class</i></p> <p>Merupakan sejenis orang, tempat, atau hal yang perlu ditangkap oleh sistem dan menyimpan informasi. Memiliki nama yang diketik dengan huruf tebal dan berpusat di kompartemen atasnya. Memiliki daftar atribut di kompartemen tengahnya. Memiliki daftar operasi di kompartemen bawahnya. Tidak secara eksplisit menunjukkan operasi yang tersedia untuk semua kelas.</p>
<p><i>Attribute name</i> <i>/derived attribute name</i></p>	<p><i>An Attribute</i></p> <p>Mewakili properti yang menggambarkan keadaan suatu objek. Dapat diturunkan dari atribut lain, ditampilkan dengan menempatkan garis miring sebelum nama atribut.</p>
<p><i>Operation name ()</i></p>	<p><i>An Operation</i></p> <p>Mewakili tindakan atau fungsi yang dapat dilakukan kelas. Dapat diklasifikasikan sebagai konstruktor, permintaan, atau operasi pembaruan. Termasuk tanda kurung yang mungkin berisi parameter atau informasi yang diperlukan untuk melakukan operasi.</p>
	<p><i>A Generalization</i></p> <p>Mewakili hubungan sejenis di antara berbagai kelas.</p>

	<p><i>An Aggregation</i></p> <p>Merepresentasikan hubungan yang logis antara beberapa kelas atau kelas dan itu sendiri. Adalah bentuk khusus dari sebuah asosiasi.</p>
---	--

Tabel II.9 Simbol-simbol *Class Diagram* (lanjutan)

Simbol	Deskripsi
<p><u><i>AssociatedWith</i></u></p> 	<p><i>An Association</i></p> <p>Mewakili hubungan antara beberapa kelas atau kelas dan itu sendiri. Diberi label menggunakan frasa kata kerja atau nama peran, yang lebih baik mewakili hubungan. Bisa ada di antara satu atau lebih kelas. Mengandung simbol multiplisitas, yang mewakili waktu minimum dan maksimum sebuah <i>instance</i> kelas dapat dikaitkan dengan <i>instance</i> kelas terkait.</p>
	<p><i>A Composition</i></p> <p>Merepresentasikan hubungan yang logis antara beberapa kelas atau kelas dan itu sendiri. Adalah bentuk khusus dari sebuah asosiasi.</p>

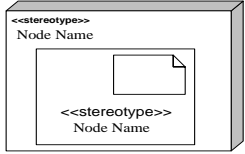
(Sumber: Dennis, 2010)

2.13.5 *Deployment Diagram*

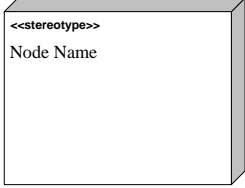
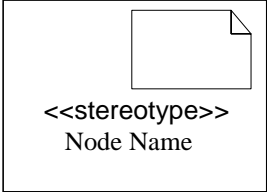
Deployment diagram digunakan untuk mewakili antara komponen perangkat keras yang digunakan dalam infrastruktur fisik dari suatu sistem informasi. Misalnya, ketika merancang sistem informasi terdistribusi yang akan menggunakan jaringan area luas, *deployment diagram* dapat digunakan untuk menunjukkan hubungan komunikasi di antara berbagai *node* dalam jaringan. Mereka juga dapat digunakan untuk mewakili komponen perangkat lunak dan bagaimana mereka dikerahkan atas arsitektur fisik ke infrastruktur sistem informasi. dalam hal ini, *deployment diagram* mewakili lingkungan untuk pelaksanaan perangkat lunak. Berikut simbol yang ada pada *deployment diagram*:

Tabel II.10 Simbol-simbol *Deployment Diagram*

Simbol	Deskripsi
--------	-----------

	<p><i>A Node with Deployed Artifact</i></p> <p>Menggambarkan artefak yang ditempatkan pada <i>node</i> fisik.</p>
---	---

Tabel II.10 Simbol-simbol *Deployment Diagram* (lanjutan)


Simbol	Deskripsi
	<p><i>A Node</i></p> <p>Adalah sumber daya komputasi, misalnya, komputer klien, <i>server</i>, jaringan terpisah atau perangkat jaringan individual. Diberi label berdasarkan nama. Dapat berisi stereotip untuk label khusus jenis <i>node</i> yang diwakili, misalnya, perangkat, <i>workstation</i> klien, <i>server</i> aplikasi, perangkat seluler, dll.</p>
	<p><i>An Artifact</i></p> <p>Adalah spesifikasi perangkat lunak atau basis data, misalnya, basis data atau tabel atau tampilan basis data, komponen atau lapisan perangkat lunak. Diberi label dengan namanya. Dapat berisi stereotip untuk jenis label khusus artefak, misalnya, file sumber, tabel basis data, file yang dapat dieksekusi, dll.</p>
<p><u><<stereotype>></u></p>	<p><i>An Communication Path</i></p> <p>Mewakili hubungan antara dua <i>node</i>. Memungkinkan <i>node</i> untuk bertukar pesan. Dapat berisi stereotip untuk secara khusus memberi label jenis jalur komunikasi yang diwakili, (misalnya, LAN, internet, serial, paralel).</p>

(Sumber: Dennis et al, 2015)




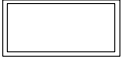
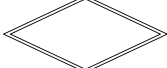
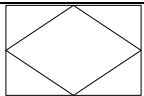

2.13.6 ERD (*Entity Relationship Diagram*)

Menurut Rosa dan Shalahuddin (2015) ERD adalah pemodelan awal basis data yang dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional, simbol-simbol yang digunakan dalam ERD, yaitu:

Tabel II.11 Simbol-simbol *Entity Relationship Diagram* (ERD)

Simbol	Deskripsi
	<i>Attribute</i> adalah properti atau karakteristik tipe entitas.

Tabel II.11 Simbol-simbol *Entity Relationship Diagram* (ERD) (lanjutan)

Simbol	Deskripsi
	<i>Strong Entity</i> adalah karakteristik atau tipe entitas yang tidak bergantung pada entitas lain.
	<i>Multivalued Attribute</i> adalah karakteristik tipe entitas yang dapat memiliki nilai lebih dari satu.
	<i>Relationship</i> adalah hubungan antar entitas.
	<i>Weak Entity</i> adalah entitas yang bergantung pada entitas lain.
	<i>Identifying Entity</i> adalah entitas yang merupakan induk dari <i>weak entity</i> .
	<i>Associative Entity</i> hanya bisa terdapat antara 2 entitas dan menggambarkan hubungan diantara 2 entitas tersebut.
	<i>Derived Attribute</i> adalah suatu atribut yang dihasilkan dari atribut yang lain.

(Sumber: Hoffer, 2010)

Di dalam ERD terdapat *multiplicity*, *multiplicity* merupakan suatu ketentuan (*constraint*) yang mengatur hubungan antara objek-objek entitas. Oleh karena itu, *multiplicity* merupakan kebijakan atau aturan bisnis (*business rule*) yang ditetapkan oleh pemakai atau organisasi, yang dapat berbeda dari satu organisasi dengan organisasi lain. Lebih lanjut, *multiplicity* dapat dibagi menjadi dua batasan atau aturan, yaitu:

1. Batasan kardinalitas (*cardinality constraint*) yang menentukan jumlah maksimum objek entitas lain yang dapat berasosiasi dengan suatu objek entitas tertentu.
2. Batasan partisipasi (*participation constraint*) yang menentukan apakah semua atau sebagian objek entitas berpartisipasi pada suatu jenis hubungan entitas.

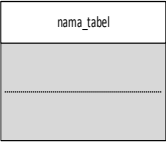
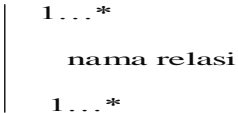
Menurut Pahlevi (2013), perbandingan kardinalitas yang dilihat dari dua arah hubungan disebut dengan rasio kardinalitas atau derajat hubungan, terdapat tiga jenis rasio kardinalitas sebagai berikut:

1. *One-to-one* (1:1), jika nilai kardinalitas pada kedua arah 1.
2. *One-to-many* (1:N) atau *many-to-one* (N:1), jika nilai kardinalitas pada salah satu arah satu (1) dan arah lainnya banyak (N).
3. *Many-to-many* (N:N), jika nilai kardinalitas pada kedua arah banyak (N).

2.13.7 *Conceptual Data Model (CDM)*

CDM (*Conceptual Data Model*) atau model konsep data merupakan konsep yang berkaitan dengan pandangan pemakai terhadap data yang disimpan dalam basis data. CDM dibuat sudah dalam bentuk tabel-tabel tanpa tipe data yang menggambarkan relasi antar tabel untuk keperluan implementasi ke basis data. CDM merupakan hasil penjabaran lebih lanjut dari ERD. Ada aturan-aturan yang harus diikuti dalam melakukan konversi ERD menjadi CDM. Berikut simbol yang ada pada *conceptual data model*:

Tabel II.12 Simbol-simbol *Conceptual Data Model (CDM)*

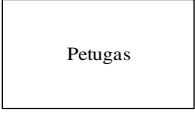
Simbol	Deskripsi
Entitas/tabel 	Entitas atau tabel yang menyimpan data dalam basis data.
Relasi 	Relasi antar tabel yang terdiri atas nama relasi dan <i>multiplicity</i> .

(Sumber: Rosa & Shalahuddin, 2015)

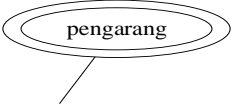
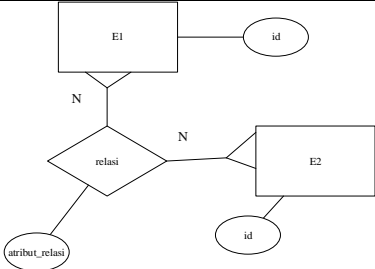
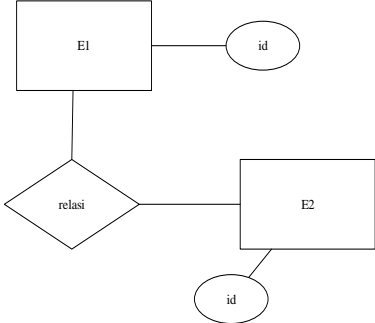
Aturan untuk mengubah ERD menjadi CDM secara umum adalah sebagai berikut:

Tabel II.13 Simbol-simbol Aturan *Conceptual Data Model (CDM)*

Simbol	Deskripsi
--------	-----------

 <p>Petugas</p> <p>Entitas</p>	E2	
	PK	<u>username</u>
		password nama
Menjadi sebuah tabel tersendiri.		

Tabel II.13 Simbol-simbol Aturan *Conceptual Data Model* (CDM) (lanjutan)

Simbol	Deskripsi											
 <p>pengarang</p> <p>Atribut <i>multivalue</i></p>	<table border="1"> <tr> <td colspan="2" style="background-color: #d9e1f2;">E2</td> </tr> <tr> <td>PK</td> <td><u>id_pustaka</u></td> </tr> <tr> <td>PK</td> <td><u>pengarang</u></td> </tr> <tr> <td></td> <td></td> </tr> <tr> <td colspan="2" style="text-align: right;"> 1..*</td> </tr> </table> <p>Menjadi sebuah tabel tersendiri dengan kunci primer (<i>primary key</i>) adalah kunci primer pada entitas dan memiliki atribut dengan nama seperti pada atribut entitas.</p>		E2		PK	<u>id_pustaka</u>	PK	<u>pengarang</u>			1..*	
E2												
PK	<u>id_pustaka</u>											
PK	<u>pengarang</u>											
1..*												
 <p>Relasi dengan kardinalitas <i>many to many</i></p>	<table border="1"> <tr> <td colspan="2" style="background-color: #d9e1f2;">E2</td> </tr> <tr> <td>PK</td> <td><u>id_E1</u></td> </tr> <tr> <td>PK</td> <td><u>id_E2</u></td> </tr> <tr> <td></td> <td></td> </tr> </table> <p>Menjadi sebuah tabel tersendiri dengan kunci primer adalah atribut yang menjadi kunci primer di kedua entitas yang direlasikannya.</p>		E2		PK	<u>id_E1</u>	PK	<u>id_E2</u>				
E2												
PK	<u>id_E1</u>											
PK	<u>id_E2</u>											
 <p>Relasi dengan kardinalitas</p>	<table border="1"> <tr> <td colspan="2" style="background-color: #d9e1f2;">E2</td> </tr> <tr> <td>PK</td> <td><u>id_E1</u></td> </tr> <tr> <td>PK</td> <td><u>id_E2</u></td> </tr> <tr> <td></td> <td></td> </tr> </table> <p>Kunci primer salah satu entitas akan dijadikan kunci asing (<i>foreign key</i>) pada tabel yang lain dan kunci asing itu dijadikan kunci primer juga,</p>		E2		PK	<u>id_E1</u>	PK	<u>id_E2</u>				
E2												
PK	<u>id_E1</u>											
PK	<u>id_E2</u>											

<i>one to one</i>	dengan kata lain, relasi tidak menjadi tabel sendiri.
-------------------	---

(Sumber: Rosa & Shalahuddin, 2015)

2.14 Kamus Data

Menurut Yakub (2012) kamus data (*data dictionary*) merupakan daftar elemen data yang teroganisir dengan definisi yang tetap dan sesuai dengan sistem sehingga *user* dan analis sistem mempunyai pengertian yang sama tentang *input*, *output*, dan data *storage*. Kamus data adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari sistem informasi.

Selain digunakan untuk dokumentasi dan mengurangi redundansi, kamus data juga dapat digunakan untuk (Yakub, 2012):

1. Memvalidasi diagram arus data dalam hal kelengkapan dan keakuratan.
2. Menyediakan suatu titik awal untuk mengembangkan layar dan laporan-laporan.
3. Menentukan muatan data disimpan dalam file-file.
4. Mengembangkan lohika untuk proses-proses diagram arus data.

2.15 Windows Navigation Diagram (WND)

Menurut Dennis et al (2015), desain struktur navigasi mendefinisikan komponen dasar antarmuka dan bagaimana mereka bekerja sama untuk menyediakan fungsionalitas kepada pengguna. *Windows Navigation Diagram* (WND) digunakan untuk menunjukkan bagaimana semua layar, bentuk, dan laporan yang digunakan oleh sistem terkait dan bagaimana pengguna berpindah dari satu ke lainnya. Kebanyakan sistem memiliki beberapa WND's, satu untuk setiap bagian utama dari sistem.

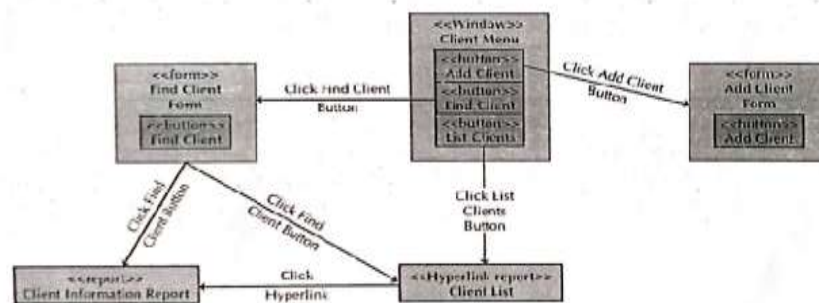
WND sangat mirip dengan *state machine*, karena keduanya memodelkan perubahan status. Mesin. Keadaannya biasanya membuat model perubahan dari suatu objek, sedangkan keadaan model perubahan lainnya yaitu di antarmuka

pengguna. Di WND setiap status antarmuka pengguna direpresentasikan sebagai kotak. Kotak biasanya sesuai dengan komponen antarmuka pengguna, seperti jendela, *form*, tombol, atau laporan.

Transisi dimodelkan sebagai panah berkepala tunggal atau berkepala dua. Panah berkepala satu menunjukkan bahwa kembali ke keadaan panggilan tidak diperlukan, sedangkan panah berkepala dua menunjukkan pengembalian yang diminta.

Item terakhir yang akan dijelaskan dalam WND adalah stereotipnya. stereotipe dimodelkan sebagai item teks yang dilampirkan dalam guillemet atau kurung sudut (<< >>). stereotipe mewakili jenis komponen antarmuka pengguna dari kotak pada diagram.

Struktur navigasi dasar antarmuka mengikuti struktur dasar dari proses bisnis itu sendiri, sebagaimana didefinisikan dalam kasus penggunaan dan model perilaku. Analisis dimulai dengan kasus-kasus penggunaan esensial dan mengembangkan aliran mendasar dari kontrol sistem ketika bergerak dari objek ke objek. Analisis kemudian memeriksa skenario penggunaan untuk melihat seberapa baik dari WND.



Gambar II.6 Contoh WND

(sumber: Dennis et al, 2015)

2.16 Personal Home Page (PHP)

Hypertext Preprocessor (PHP) adalah bahasa *server-side scripting* yang digunakan untuk aplikasi web yang dinamis dan interaktif. Sebuah halaman PHP adalah sebuah halaman HTML yang memiliki *server-side scripts* yang

ditempatkan dalam *server* dan diproses oleh *web server* sebelum dikirim ke *browser* pemakai (Welling dan Thomson, 2003).

Server-side scripts dijalankan ketika *browser* melakukan permintaan *file.php* dari *server*. PHP dipanggil oleh *web server*, dimana proses script perintah yang ada di suatu halaman dieksekusi mulai dari awal sampai akhir di dalam mesin PHP. Setelah *script* PHP tersebut diolah, hasilnya akan ditampilkan kepada *client* melalui *web browser* berupa tampilan HTML.

Menurut Anhar (2010) beberapa kelebihan PHP dari bahasa pemrograman lain adalah sebagai berikut:

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. *Web server* yang mendukung PHP dapat ditemukan dimana-mana dari mulai *apache*, *ISS*, *Lighttpd*, hingga *Xitami* dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyak milis-milis dan *developer* yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
5. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin seperti *Linux*, *Unix*, *Macintos*, dan *Windows* serta secara *runtime console* dan dapat menjalankan perintah-perintah system

2.17 MySQL

Menurut Anhar (2010), MySQL adalah sebuah perangkat lunak sistem manajemen basis data (DBMS) dari sekian banyak DBMS seperti Oracle, MySQL, Postgre SQL, dan lain-lain. MySQL merupakan DBMS yang *multithread*, *multi-user* yang bersifat gratis dibawah GNU *General Public License* (GPL). Lisensi GPL ini membebaskan pengguna menggunakan MySQL tanpa membayar royalti kepada MySQL AB singkatan "AB" di belakang MySQL adalah singkatan dari "Akiebolag" istilah PT bagi perusahaan Swedia. Tahun 2005 MySQL AB merilis MySQL versi 5 dengan model bisnis dan aplikasi yang

sudah lengkap. Pada tahun 2008 *Sun Microsystems, Inc* mengumumkan akuisisi terhadap MySQL AB seharga US\$1 Milyar (MySQL, 2011). Selang setahun di bulan April 2009 *Oracle Corp* memasuki sebuah kesepakatan untuk membeli *Sun Microsystem*, pemilik dari MySQL hak cipta dan merk dagang, dengan pimpinan *Sun Microsystem* menyetujui kesepakatan tersebut. Pada tanggal 14 desember 2009 *Oracle* berjanji untuk terus meningkatkan MySQL seperti yang telah dilakukan selama empat tahun sebelumnya. Sebuah gerakan melawan akuisisi *Oracle* dari MySQL, untuk “Save MySQL” dari *Oracle* dimulai oleh salah satu penemu MySQL yaitu Monty Widenius. Namun, pada akhirnya MySQL berhasil di akuisisi oleh *Oracle* pada tanggal 21 januari 2010. Transaksi yang terjadi antara *Oracle* dengan *Sun Microsystem* bernilai kira-kira US\$ 7,4 miliar, atau US\$ 5,6 miliar bersih dari uang tunai dan hutang *Sun Microsystem* (*Business Wire*, 2009).

MySQL mendukung banyak tipe data yang dapat disimpan pada sebuah kolom. Penjelasan singkat kegunaan masing-masing jenis data akan sangat bermanfaat dalam memilih suatu jenis data yang dipakai dalam merancang tabel. Berikut jenis data yang tersedia pada MySQL:

Tabel II.14 Jenis Data pada MySQL

Jenis Data	Keterangan
CHAR	Sebuah <i>string</i> dengan panjang tetap. Sisa jumlah karakter yang belum terisi akan diisi dengan spasi, akan tetapi spasi ini dibuang jika data dipanggil. Jangkauan nilai M adalah 1-255 karakter.
VARCHAR	<i>String</i> dengan panjang berupa variabel M bisa mencapai 65535.
DATE	Data berupa tanggal. Format tanggal dalam bentuk ‘YYYY-MM-DD’.
TIME	Data berupa waktu. Format waktu dalam bentuk ‘HH:MM:SS’.
TINYINT	Bilangan antara -128 sampai dengan +127.
SMALLINT	Bilangan antara -32768 sampai dengan +32767.
INT	Bilangan antara -2147683648 sampai dengan +2147683647

FLOAT	Bilangan <i>floating point</i> yang kecil (presisi tunggal). Jangkauan nilai yang diizinkan adalah -3.402823466E+38 s/d -1.175494351e-38,0 dan 1.175494351E-38 s/d 3.402823466E+38.
TEXT, BLOB	Sebuah TEXT atau BLOB dengan panjang karakter maksimum 65535 karakter.
DOUBLE	Bilangan <i>floating point</i> dengan ukuran normal (presisi ganda). Jangkauan nilai yang diizinkan adalah -1.7976931348623157E+308 s/d -2.225073858507201E-308,0 dan 2.225073858507201E-308 s/d 1.7976931348623157E+308.

Tabel II.14 Jenis Data pada MySQL (lanjutan)

Jenis Data	Keterangan
ENUM	Sebuah <i>enumeration</i> . Sebuah objek <i>string</i> yang hanya boleh memiliki satu nilai, yang terambil dari 'value1', 'value2', ..., NULL atau nilai spesial "" <i>error</i> . Sebuah <i>enum</i> dapat menampung 65535 pilihan nilai.

(Sumber: Sutaji, 2012)

2.18 XAMPP

XAMPP adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolahan data MySQL di komputer lokal. XAMPP berperan sebagai *web server* pada komputer. XAMPP juga dapat disebut sebagai sebuah *CPanel server virtual*, yang dapat membantu melakukan *preview* sehingga dapat memodifikasi *website* tanpa harus *online* atau terakses internet (Kadir, 2014).

Fungsi lainnya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP Server, MySQL *database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X, Apache, MySQL, PHP dan Perl. Program ini

tersedia dalam GNU *General Public License* dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis.

MySQL mendukung banyak tipe data yang dapat disimpan pada sebuah kolom. Penjelasan singkat kegunaan masing-masing jenis data akan sangat bermanfaat dalam memilih suatu jenis data yang dipakai dalam merancang tabel (Sutaji, 2012).

BAB III

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Metodologi penelitian atau *methodology of research* berasal dari kata “*metoda*” yang berarti cara atau teknik dan “*logos*” yang berarti ilmu. Untuk mendapatkan gambaran yang lebih jelas tentang arti metodologi penelitian, maka beberapa pengertian atau definisi metodologi penelitian yang dikutip dari buku Soewardi (2012), yaitu:

1. Menurut Komarudin metodologi penelitian diartikan sebagai suatu proses mencari jawaban terhadap suatu pertanyaan atau masalah melalui prosedur yang sistematis dan terawasi.
2. Menurut Sutrisno Hadi, MA pengertian metodologi penelitian lebih menekankan pada suatu usaha untuk menemukan, mengembangkan dan menguji kebenaran suatu pengetahuan, dimana dilakukan dengan menggunakan metoda-metoda ilmiah.

Dari beberapa pengertian dan definisi yang tersebut diatas maka dapat diperoleh kesimpulan pengertian metodologi penelitian adalah sebagai suatu usaha atau proses mencari jawaban atas suatu pertanyaan atau masalah dengan cara sabar, hati-hati, terencana, sistematis atas fakta atau prinsip-prinsip, mengembangkan dan menguji kebenaran ilmiah suatu pengetahuan (Soewardi, 2012).

3.2 Jenis dan Sumber Data

Data yang diperlukan dalam penelitian ini ada dua jenis, yaitu:

1. Data Primer

Data Primer adalah data yang diperoleh secara langsung dari PT Sukses Cipta Makmur, dimana pengumpulan data atau informasi dilakukan langsung dari objek yang diteliti. Data-data tersebut adalah data yang digunakan dalam proses pengelolaan surat perintah kerja diantaranya

struktur organisasi, analisis sistem yang telah berjalan, proses bisnis sistem saat ini dan yang akan diusulkan, dan kebutuhan pengguna sistem.

2. **Data Sekunder**

Data sekunder adalah data yang diperoleh dari data yang tersedia dan telah terlebih dahulu dikumpulkan dan dilaporkan oleh orang lain, buku-buku dan kajian ilmiah dari berbagai sumber yang berkaitan dengan penelitian. Data tersebut adalah data umum perusahaan, profil perusahaan, dan struktur organisasi perusahaan.

3.3 Metode Pengumpulan Data

Menurut Sugiyono (2013) teknik pengumpulan data merupakan langkah yang paling strategis dalam penelitian, karena tujuan utama dari penelitian adalah mendapatkan data. Penelitian ini menggunakan beberapa teknik pengumpulan data, yaitu:

1. **Studi Lapangan**

Studi lapangan adalah tahap melakukan pengamatan langsung terhadap pengguna sistem dan pengamatan langsung terhadap sistem yang sedang berjalan di PT Sukses Cipta Makmur. Observasi ini dilakukan pada divisi PPIC untuk mengetahui alur proses kegiatan pengelolaan surat perintah kerja.

2. **Wawancara**

Wawancara yaitu mencari data yang dibutuhkan secara langsung melalui memberikan pertanyaan spontan terhadap segala hal yang diperlukan pada penyusunan Tugas Akhir ini. Wawancara ini dilakukan kepada karyawan pada divisi PPIC PT Sukses Cipta Makmur mengenai alur kegiatan pengelolaan surat perintah kerja.

3. **Studi Kepustakaan**

Studi kepustakaan dilakukan dengan membaca buku dan literatur dalam lingkup perkuliahan maupun di luar lingkup perkuliahan yang

berhubungan dengan judul dan permasalahan, sehingga dapat menunjang dalam penulisan tugas akhir ini. Studi kepustakaan yang dilakukan adalah dengan menggunakan buku yang dimiliki, buku yang dipinjam dari perpustakaan dan mencari data yang diperlukan melalui Internet.

3.4 Metode Pengembangan Sistem

Pada penelitian ini untuk mengatasi masalah yang ada pada sistem, diputuskan untuk membuat pengembangan sistem. Sistem dikembangkan menjadi lebih terkomputerisasi, terintegrasi dan berbasis *database*. Pengembangan sistem dilakukan dengan menggunakan metode *evolutionary prototype*. Metode prototipe ini terdiri dari tahap identifikasi kebutuhan pengguna (*user*), membuat prototipe, menentukan prototipe diterima atau tidak dan menggunakan prototipe. Berikut ini akan dibahas secara singkat mengenai tahapan-tahapan yang dilakukan dalam pengembangan sistem.

a. Mengidentifikasi kebutuhan *user*

Pengembang mewawancarai karyawan PPIC untuk mendapatkan ide mengenai apa yang diminta dari sistem yang diinginkan.

b. Membuat *prototype*

Pengembang membuat *prototype* dari sistem yang telah dijelaskan agar lebih mendapatkan bayangan apa yang sebenarnya diinginkan.

c. Menyesuaikan dan evaluasi *prototype* dengan keinginan *user*

Pengembang menanyakan kepada *user* tentang *prototype* yang sudah dibuat, apakah sesuai atau tidak dengan kebutuhan sistem. Jika ya, langkah selanjutnya akan diambil. Jika tidak, *prototype* direvisi dengan mengulang kebutuhan *user*.

d. Menggunakan *prototype*

Sistem mulai dikembangkan dengan *prototype* yang sudah dibuat

3.5 Kerangka Penelitian

Dalam penelitian yang dibahas dalam tugas akhir ini, dilakukan langkah-langkah atau tahapan dalam penelitian yang ada pada Gambar III.1 sesuai dengan metodologi penelitian yang digunakan dalam penelitian tersebut. Penjelasan langkah-langkah atau tahapan dalam penelitian tersebut adalah sebagai berikut:

1. Studi Pendahuluan

Pada tahap ini, merupakan langkah awal yang dilakukan untuk memulai penelitian. Dengan menggunakan tiga metode yaitu observasi (pengamatan), wawancara dan studi pustaka.

2. Pokok Permasalahan

Pada tahap ini, bertujuan untuk mengetahui sistem pengelolaan surat perintah kerja yang sedang berjalan. Serta mengetahui masalah yang ada pada sistem tersebut.

3. Tujuan Penelitian

Pada tahap ini, dimaksudkan agar dapat memberikan masukan atau jalan keluar dari pokok permasalahan.

4. Batasan Masalah

Pada tahap ini, bertujuan untuk membatasi masalah agar penelitian lebih terarah sehingga dapat menunjukkan gambaran yang lebih spesifik.

5. Pengembangan Sistem Menggunakan *Evolutionary Prototype*

- a. Identifikasi Kebutuhan *User*

Pada tahap ini, mengidentifikasi apa saja yang dibutuhkan oleh pengguna sistem informasi pengelolaan surat perintah kerja. Memilih metode pengembangan sistem yaitu *evolutionary prototype*. Cara untuk mendapatkan informasi kebutuhan *user* adalah sebagai berikut:

- 1) Pengumpulan Data

Dalam mengumpulkan data, metode yang digunakan yaitu dengan metode observasi, wawancara dan studi pustaka. Metode wawancara dilakukan dengan kegiatan tanya jawab terhadap karyawan PPIC yang terkait dengan kegiatan pengelolaan surat

perintah kerja. Pengamatan dilakukan dengan cara sebagai berikut:

- a) Mempelajari dokumen perusahaan, seperti profil perusahaan dan bagan struktur organisasi.
- b) Mempelajari sistem informasi pengelolaan surat perintah kerja yang sedang berjalan di PT Sukses Cipta Makmur, termasuk aliran dokumen yang berhubungan dengan bahan baku masuk, keluar dan sisa bahan baku serta terselesainya perintah kerja.

2) Pengolahan Data

Data-data yang didapatkan kemudian diolah untuk menganalisis sistem yang sedang berjalan guna mengetahui kekurangan yang terdapat didalam sistem tersebut untuk kemudian dijadikan bahan pertimbangan dalam merancang dan membuat sistem informasi pengelolaan surat perintah kerja sesuai dengan kebutuhan *user*.

b. Pembuatan *prototype*

Membuat *prototype* sesuai dengan analisis dan perancangan yang dilakukan dengan cara sebagai berikut:

- 1) Memodelkan sistem menggunakan *Unified Modeling Language* (UML), *use case diagram*, *activity diagram* dan *sequence diagram*.
- 2) Memodelkan data menggunakan *class diagram* dan kamus data.
- 3) Merancang sistem dengan *flowchart*, *Windows Navigation Diagram* (WND) dan tampilan antar muka.
- 4) Pembuatan *prototype* dengan bahasa pemrograman PHP 5.6.12 dan *database* MySQL 5.6.26.

c. Menyesuaikan dan Evaluasi *prototype* dengan Keinginan *User*

Pada tahap ini, *prototype* yang diusulkan akan diperlihatkan dan diuji coba terlebih dahulu kepada *user* apakah sudah sesuai dengan keinginan *user* atau belum. Jika belum maka dilakukan analisis

kembali, tetapi apabila sudah sesuai maka tahap selanjutnya adalah implementasi *prototype* tersebut kepada *user*.

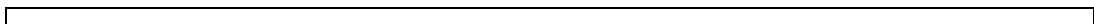
d. Menggunakan *prototype*

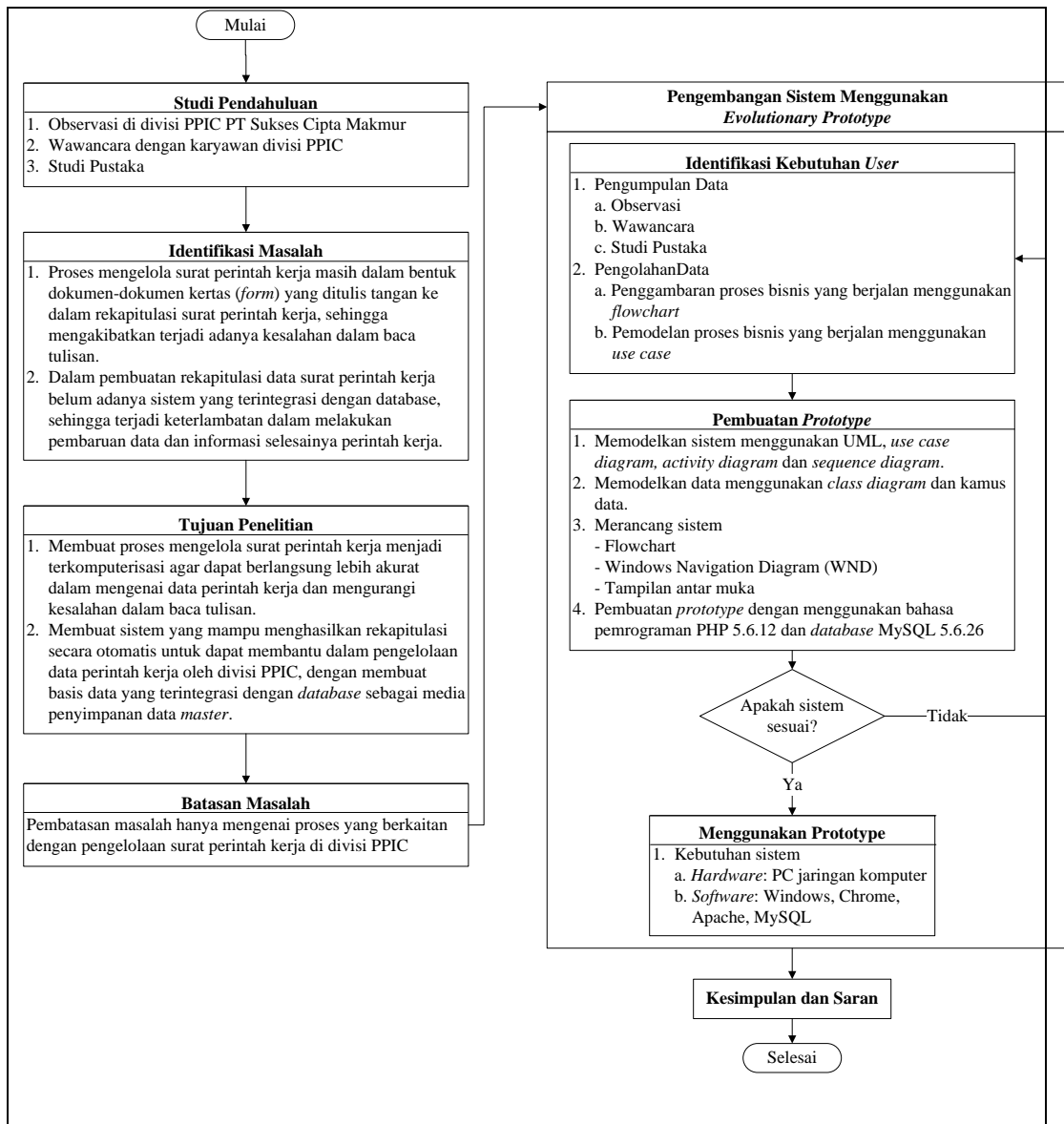
Pada tahap ini, *prototype* dari sistem yang penulis usulkan akan dilihat apakah sistem usulan tersebut sudah sesuai dengan keinginan karyawan PPIC atau belum. Jika belum, maka penulis akan melakukan pengumpulan data kembali, menganalisis data, mencari tahu kebutuhan apa lagi yang dibutuhkan oleh *user* dari sistem, serta melakukan perancangan sistem ulang. Sedangkan jika sudah sesuai keinginan dengan karyawan PPIC, maka *prototype* dapat digunakan.

6. Kesimpulan dan Saran

Mengambil kesimpulan dari hasil analisis sistem yang berjalan sistem yang diusulkan dan sistem yang dirancang serta memberikan saran.

Kerangka penelitian dibuat dalam bentuk *flowchart* yang menggambarkan tahap-tahap kegiatan mulai dari awal hingga akhir dapat dilihat pada Gambar III.1.





Gambar III.1 Kerangka Penelitian

(Sumber: Pengolahan Data, 2018)

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil pengumpulan, pengolahan dan analisis data yang telah dilakukan dalam penelitian ini dapat disimpulkan bahwa:

1. Dengan aplikasi sistem ini dapat membantu proses mengelola surat perintah kerja menjadi terkomputerisasi agar dapat berlangsung lebih akurat dalam mengenai data perintah kerja dan mengurangi salah baca tulisan.
2. Dengan aplikasi sistem ini dapat membantu menyediakan rekapitulasi secara otomatis dan informasi tentang perintah kerja yang telah terselesaikan.

6.2 Saran

Saran yang diberikan untuk pengembangan sistem informasi pengelolaan surat perintah kerja ini selanjutnya adalah sebagai berikut:

1. Untuk penelitian selanjutnya diharapkan adanya notifikasi setiap perintah kerja selesai tersimpan sehingga divisi produksi tidak perlu mencari perintah kerja.
2. Untuk penerapan sistem baru terhadap sistem lama, sebaiknya dilakukan secara bertahap dan diperlukan sosialisasi penggunaan sistem ini kepada bagian yang terkait.
3. Sebaiknya dilakukan pemeliharaan aplikasi secara berkala, sehingga aplikasi dapat berjalan dengan baik.

DAFTAR PUSTAKA

- Adisaputro, Gunawan., dan Marwan Asri. 2016. *Anggaran Perusahaan*. Edisi Kedua, Cetakan Kedelapan. Yogyakarta: BPFE.
- Al Fatta, Hanif. 2007. *Analisis dan Perancangan Sistem Informasi untuk Keunggulan Bersaing Perusahaan dan Organisasi Modern*. Yogyakarta: Andi Offset.
- Ashby, Michael. 2005. *Materials & Design*. Edisi Terjemahan. Italy: Elsevier Butterwath Hamemann.
- Andika, Dwiki. 2016. *Flowchart*. <https://www.it-jurnal.com/pengertian-flowchart/> (Tanggal Akses: 24 April 2018).
- Anhar. 2010. *Panduan Menguasai PHP & MySQL Secara Otodidak*. Cetakan Pertama. Jakarta: Mediakita.
- Dennis, Alan., Barbara Haley., dan David Tegarden. 2015. *System Analysis and Design*. Fifth Edition. America: Willey.inc.
- Fauzi, A. 2007. *Mail Marge: Solusi Praktis untuk Surat-Surat Bisnis*. Jakarta: Elex Media Komputindo.
- Herjanto, Eddy. 2008. *Manajemen Operasi*. Edisi Ketiga. Jakarta: Grasindo.
- Hutahaean, Jeperson. 2014. *Konsep Sistem Informasi*. Edisi Pertama, Cetakan Pertama. Yogyakarta: Deepublish.
- Jogiyanto, H.M. 2005. *Analisa dan Desain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktik Aplikasi Bisnis*. Yogyakarta: Andi.
- Kadir, Abdul. 2014. *Pengenalan Sistem Informasi Edisi Revisi*. Yogyakarta: Andi.
- McLeod, Raymond., dan Gerge. 2012. *Sistem Informasi Manajemen*. Jakarta: Salemba Empat.
- Mulyadi. 2010. *Sistem Akuntansi*. Edisi Ketiga, Cetakan Kelima. Jakarta: Salemba Empat.
- Pratama, I Putu Agus Eka. 2016. *Integrasi & Migrasi Sistem Teori dan Praktik*. Bandung: Informatika.
- Purwanto, Djoko. 2006. *Komunikasi Bisnis*. Edisi ketiga. Jakarta: Erlangga.

Rosa, A. S., dan M. Shalahuddin. 2016. *Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Bandung: Informatika..

Saifuddin. 2014. *Pengelolaan Pembelajaran Teoretis dan Praktis*. Yogyakarta: Deepublish.

Sutabri, Tata. 2012. *Analisis Sistem Informasi*. Yogyakarta: Andi.

Sutaji, Deni. 2012. *Sistem Inventory Mini Market dengan PHP dan Jquery*. Yogyakarta: Lokomedia.

Welling, L., dan Thomson, L. 2003. *PHP and MySQL Web Development, Second Edition*. Indianapolis: Sams Publishing.

Yakub. 2012. *Pengantar Sistem Informasi*. Yogyakarta: Graha Ilmu.

Satzinger, 2012

Hoffer, 2010 *Entity Relationship Diagram*

Pahlevi, 2013

Richardson dan Thies. 2013 *.Secure Software Design*

Situs:

Anonim. 2017. *Flowchart*. https://id.wikipedia.org/wiki/Diagram_alir (Tanggal Akses: 24 April 2018).

Anonim. 2009. Mysql. <https://www.businesswire.com/news/home/20090420005760/en/Oracle-Buy-Sun> (Tanggal Akses: 24 April 2018).

Anonim. 2010. Mysql. <https://web.archive.org/web/20110718044718/http://mysql.com/news-and-events/sun-to-acquire-mysql.html> (Tanggal Akses: 24 April 2018).

Anonim. 2018. Sistem. <https://id.wikipedia.org/wiki/Sistem> (Tanggal Akses: 24 April 2018).

LAMPIRAN A

TAMPILAN PROGRAM

1. Halaman Login

Please Sign In

2. Halaman Utama

PT Sukses Cipta Makmur

Dashboard

Selamat datang admin di Website PT Sukses Cipta Makmur

3. Data Master

PT Sukses Cipta Makmur

Bahan Baku

Data Bahan Baku

No.	Id Bahan Baku	Nama Bahan Baku	Diameter	Panjang awal	Panjang akhir	Rencana Cutting	
1	BB001	HWS (14)	14	6000	600	28	
2	BB002	S15 (12)	12	3000	600	24	
3	BB003	S20 (11)	11	3000	600	26	
4	BB004	S20 (23 S)	24	3000	600	28	
5	BB005	S45 (8)	8	6000	600	33	
6	BB006	ST41 (8)	8	6000	600	30	

Jumlah Data: 6

PT Sukses Cipta Makmur

Dashboard

Data Master

- Data Bahan Baku
- Data Spare Part
- Data Customer
- Perintah Kerja
- Perencanaan Bahan Baku
- Pengiriman Bahan Baku
- Pemantauan Bahan Baku Mentah
- Laporan

Spare Part

Data Spare Part

Pencarian

No.	ID Spare Part	Nama Spare Part	Proses	D
1	SP001	BOLT REAR K29	onc, grinding, packing, gudang	0 0
2	SP002	COLLAR FR DIST 37	turnet, planing, packing, gudang	0 0
3	SP003	RPE B	turnet, milling, packing, gudang	0 0
4	SP004	RPE INNER BUSHING HANGER	turnet, grinding, planing, packing, gudang	0 0
5	SP005	RPE OUTER BUSHING HANGER	onc, milling, packing, gudang	0 0
6	SP006	PVOT BRAKE PEDAL	onc, turnet, grinding, packing, gudang	0 0

Jumlah Data: 6

PT Sukses Cipta Makmur

Dashboard

Data Master

- Data Bahan Baku
- Data Spare Part
- Data Customer
- Perintah Kerja
- Perencanaan Bahan Baku
- Pengiriman Bahan Baku
- Pemantauan Bahan Baku Mentah
- Laporan

Customer

Data Customer

Pencarian

No.	ID Customer	Nama Customer	Alamat	Email	D
1	CU001	PT Adri Wijaya Cika	Jl Raya Negerang Km 12, Banjar Gedang	adri@awparts.com	0 0
2	CU002	PT Agratama Multi Agung	J Pangeran Semang, Des. Sekeloa, Geger	agratama.multiaugur@gmail.com	0 0
3	CU003	PT Astra Cipta	Jl Raya Pegunungan Dua Kelapa Gedang	contact@component.astra.co.id	0 0
4	CU004	PT Cipta Nugraha Cipta	Delta Solcon Industrial Park, Lippo Cikarang	onc@certim.net.id	0 0
5	CU005	PT Nanjaya Karya Perkasa	J KH. Liliat, Rp Raya Kar Dayan Cikarang	nanjaya@nanjaya-karya-perkasa.com	0 0

Jumlah Data: 5

4. Data Transaksi

PT Sukses Cipta Makmur

Dashboard

Data Master

- Perintah Kerja
- Perencanaan Bahan Baku
- Pengiriman Bahan Baku
- Pemantauan Bahan Baku Mentah
- Laporan

Perintah Kerja

Data Perintah Kerja

Pencarian

No.	Noor PK	Tgl. Order	Nama Customer	ID Customer	Nama Spare Part	ID Spare Part	Proses	Total PO	Rencana Bahan Baku	ID Bahan Baku	Total BB	Tgl. Delivery	Status QC	Status Prodikal	D
1	PK001	2019-08-15	PT Adri Wijaya Cika	CU001	BOLT REAR K29	SP001	onc, grinding, packing, gudang	1000	1000	BB001	1000	2019-08-25	Tervalidasi	Tervalidasi	0 0
2	PK002	2019-08-15	PT Agratama Multi Agung	CU002	COLLAR FR DIST 37	SP002	turnet, planing, packing, gudang	1000	1000	BB002	1000	2019-08-25	Tervalidasi	Tervalidasi	0 0
3	PK003	2019-08-16	PT Astra Cipta	CU003	RPE B	SP003	turnet, milling, packing, gudang	1000	1000	BB003	1000	2019-09-31	Tervalidasi	Tervalidasi	0 0

Jumlah Data: 3

PT Sukses Cipta Makmur

- Dashboard
- Data Master
- Perintah Kerja
- Permintaan Bahan Baku
- Pengiriman Bahan Baku
- Pemotongan Bahan Baku Mentah
- Laporan

Permintaan Bahan Baku

Data Permintaan Bahan Baku

Pencarian 🔍 [Tambah Permintaan](#)

No.	Nomor Permintaan	Tgl Permintaan	Nomor PK	Nama Bahan Baku	Id Bahan Baku	Total Bahan Baku	Jumlah Hari	Total Hari	Status PPIC	
1	PER001	2018-09-18	PK001	S15 (12)	SP001	1000	100	10	Tersedia	
2	PER002	2018-09-15	PK002	Hexa (14)	SP002	1000	100	10	Tersedia	
3	PER003	2018-09-04	PK003	ST41 (8)	SP003	1000	200	5	Tersedia	

Jumlah Data: 3

PT Sukses Cipta Makmur

- Dashboard
- Data Master
- Perintah Kerja
- Permintaan Bahan Baku
- Pengiriman Bahan Baku
- Pemotongan Bahan Baku Mentah
- Laporan

Pengiriman Bahan Baku

Data Pengiriman Bahan Baku

Pencarian 🔍 [Tambah Pengiriman](#)

No.	Nomor Pengiriman	Nomor PK	Nama Bahan Baku	Id Bahan Baku	Total Kirim	Tgl Kirim	Status Produksi	
1	PENG001	PK001	S15 (12)	SP002	100	2018-09-04	Tersedia	
2	PENG002	PK002	Hexa (14)	SP001	100	2018-09-04	Tersedia	
3	PENG003	PK003	ST41 (8)	SP000	100	2018-09-04	Tersedia	
4	PENG004	PK001	S15 (12)	SP003	100	2018-09-05	Tersedia	

Jumlah Data: 4

PT Sukses Cipta Makmur

- Dashboard
- Data Master
- Perintah Kerja
- Permintaan Bahan Baku
- Pengiriman Bahan Baku
- Pemotongan Bahan Baku Mentah
- Laporan

Pemotongan Bahan Baku Mentah

Data Pemotongan bahan baku mentah

Pencarian 🔍 [Tambah Pemotongan](#)

No.	Nomor Pemotongan	Tgl Pemotongan	Nama Bahan Baku	Id Bahan Baku	Bencana Cutting	Panjang Cutting	Panjang Awal Bahan Baku	Aktual Cutting	
1	PEM001	2018-09-04	Hexa (14)	BB001	28	600	6000	280	
2	PEM002	2018-09-04	S15 (12)	BB002	24	600	3000	120	
3	PEM003	2018-09-03	S20 (17)	BB003	26	600	3000	130	
4	PEM004	2018-09-04	S25 (21.5)	BB004	28	600	3000	140	
5	PEM005	2018-09-04	S45 (8)	BB005	32	600	3000	220	

5. Laporan

PT Sukses Cipta Makmur

Dashboard

Data Master

Perintah Kerja

Pembelian Bahan Baku

Pengiriman Bahan Baku

Pembagian Bahan Baku Mentah

Laporan

Ketersediaan Bahan Baku

Rekapitulasi Perintah Kerja

Ketersediaan Bahan Baku

Data Ketersediaan Bahan Baku

Cetak

Apa Bahan Baku Yang Kurang Dari Minimum Bole

No.	Nama Bahan Baku	Total Bahan Baku Keluar	Total Bahan Baku Masuk	Total Ketersediaan Bahan Baku
1	BB001	100	200	100
2	BB002	1000	1000	0
3	BB003		100	100
4	BB004		200	200
5	BB005		200	200
6	BB006	100	200	100

PT Sukses Cipta Makmur

Dashboard

Data Master

Perintah Kerja

Pembelian Bahan Baku

Pengiriman Bahan Baku

Pembagian Bahan Baku Mentah

Laporan

Ketersediaan Bahan Baku

Rekapitulasi Perintah Kerja

Rekapitulasi Perintah Kerja

Data Rekapitulasi Perintah Kerja

Cetak

No.	Nomer PK	Nama Customer	Nama Spare Part	Nama Bahan Baku	Total Kebutuhan Bahan Baku	Total Bahan Baku Keluar	Tanggal	Status
1	PK001	CU001	SP001	BB001	1000	1000	2018-09-05	Selesai
2	PK002	CU002	SP002	BB002	1000	100	2018-09-04	Belum Selesai
3	PK003	CU003	SP003	BB003	1000	100	2018-09-04	Belum Selesai

Jumlah Data 3

6. Cetak Laporan

9/16/2018

localhost/td_pk/rekapitulasi_laporan_bulanan.php

REKAPITULASI PERINTAH KERJA PT SUKSES CIPTA MAKMUR

Tanggal Cetak : 2018-09-16

No.	Nomer PK	Nama Customer	Nama Spare Part	Nama Bahan Baku	Total Kebutuhan Bahan Baku	Total Bahan Baku Keluar	Tanggal	Status
1	PK001	CU001	SP001	BB001	1000	1000	2018-09-05	Selesai
2	PK002	CU002	SP002	BB002	1000	100	2018-09-04	Belum Selesai
3	PK003	CU003	SP003	BB003	1000	100	2018-09-04	Belum Selesai

LAMPIRAN B

KODE PROGRAM

7. Config

```
<?php session_start();
//setting default timezone
date_default_timezone_get('Asia/Jakarta');
//koneksi
$con = mysqli_connect('localhost', 'root', '', 'permintaan');
if(mysqli_connect_errno()) {
    echo mysqli_connect_error();
}
//fungsi base_url
function base_url($url = null) {
    $base_url = "http://localhost/cb_pk";
    if($url !=null) {
        return $base_url."/".$url;
    } else {
        return $base_url;
    }
}
}?>
```

8. Login

```
<?php require_once "../_config/config.php";
if(isset($_SESSION['user'])) {
    //echo
"<script>>window.location='".$base_url()."/dashboard/index.html';</script>";
} else {
?>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="description" content="">
  <meta name="author" content="">
  <title>Login - Pt Sukses Cipta Makmur</title>
  <!-- Bootstrap Core CSS -->
  <link href="../vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <!-- MetisMenu CSS -->
  <link href="../vendor/metisMenu/metisMenu.min.css" rel="stylesheet">
  <!-- Custom CSS -->
  <link href="../dist/css/sb-admin-2.css" rel="stylesheet">
  <link rel="icon" href="<?=base_url('logo.png')?>">
  <!-- Custom Fonts -->
  <link href="../vendor/font-awesome/css/font-awesome.min.css"
rel="stylesheet" type="text/css">
  <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and
media queries -->
  <!-- WARNING: Respond.js doesn't work if you view the page via file:// --
>
  <!--[if lt IE 9]>
    <script
src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
    <script
src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>
  <![endif]-->
</head>
<body>
```

```

<div class="container">
  <div class="row">
    <div class="col-md-4 col-md-offset-4">
      <div class="login-panel panel panel-default">
        <div class="panel-heading">
          <h3 class="panel-title">Please Sign In</h3>
        </div>
        <div class="panel-body">
          <?php
            if(isset($_POST['login'])) {
              $user      =      mysqli_real_escape_string($con,
trim($_POST['user']));
              $pass      =      sha1(mysqli_real_escape_string($con,
trim($_POST['pass'])));
              $sql_login = mysqli_query($con, "SELECT * FROM
tb_user WHERE username = '$user' AND password = '$pass'") or die
(mysqli_error($con));
              if(mysqli_num_rows($sql_login) > 0) {
                $_SESSION['user'] = $user;
                echo
" <script>window.location="" .base_url()."/dashboard/";</script>";
              } else { ?>
            }
          </div>
        </div>
        <div align="center" class="alert alert-danger alert-
dismissable" role="alert">
          <a href="#" class="close" data-dismiss="alert" aria-
label="close">&times;</a>
          <span class="glyphicon glyphicon-exclamation-sign"
aria-hidden="true"></span>
          <strong>Login gagal!</strong> Username / password
salah

```

```

        </div>
    </div>
    <?php
    }
    } ?>
    <form action="" method="post" role="form">
        <fieldset>
            <div class="form-group">
                <input type="text" name="user" class="form-control"
placeholder="Username" required autofocus>
            </div>
            <div class="form-group">
                <input type="password" name="pass" class="form-
control" placeholder="Password" required>
            </div>
            <div class="input-group">
                <input type="submit" name="login" class="btn btn-lg
btn-success btn-block" value="Login">
            </div>
        </fieldset>
    </form>
    </div>
</div>
</div>
</div>
</div>
<!-- jQuery -->
<script src="../../vendor/jquery/jquery.min.js"></script>
<!-- Bootstrap Core JavaScript -->
<script src="../../vendor/bootstrap/js/bootstrap.min.js"></script>
<!-- Metis Menu Plugin JavaScript -->
<script src="../../vendor/metisMenu/metisMenu.min.js"></script>

```

```
<!-- Custom Theme JavaScript -->
<script src="../../dist/js/sb-admin-2.js"></script>
</body>
</html>
<?php
} ?>
```

9. Index

```
<?php require_once "_config/config.php";
if(isset($_SESSION['user'])) {
    echo "<script>>window.location='".base_url('dashboard')."';</script>";
} else {
    echo "<script>>window.location='".base_url('auth/login.php')."';</script>";
} ?>
```

10. Header

```
<?php require_once "_config/config.php";
if(isset($_SESSION['user'])) {
    //echo
"<script>>window.location='".base_url('auth/login.php')."';</script>";
} ?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">
    <title>Sistem Pengelolaan Surat Perintah Kerja</title>
    <link rel="icon" href="<?=base_url('logo.png')?>">
```

```
<!-- Bootstrap Core CSS -->
<link href="../../vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
<!-- MetisMenu CSS -->
<link href="../../vendor/metisMenu/metisMenu.min.css" rel="stylesheet">
<!-- Custom CSS -->
<link href="../../dist/css/sb-admin-2.css" rel="stylesheet">
<!-- Morris Charts CSS -->
<link href="../../vendor/morrisjs/morris.css" rel="stylesheet">
<!-- Custom Fonts -->
<link href="../../vendor/font-awesome/css/font-awesome.min.css"
rel="stylesheet" type="text/css">
<!-- jQuery -->
<script src="../../vendor/jquery/jquery.js"></script>
<!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and
media queries -->
<!-- WARNING: Respond.js doesn't work if you view the page via file:// --
>
<!--[if lt IE 9]>
<script
src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
<script
src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>
<![endif]-->
</head>
<body>
<div id="wrapper">
<!-- Navigation -->
<nav class="navbar navbar-default navbar-static-top" role="navigation"
style="margin-bottom: 0">
<div class="navbar-header">
```

```

        <button type="button" class="navbar-toggle" data-
toggle="collapse" data-target=".navbar-collapse">
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
    </button>
    <a class="navbar-brand" href="index.html">PT Sukses Cipta
Makmur</a>
</div>
<!-- /.navbar-header -->
<ul class="nav navbar-top-links navbar-right">
    <!-- /.dropdown -->
    <li class="dropdown">
        <a class="dropdown-toggle" data-toggle="dropdown" href="#">
            <i class="fa fa-user fa-fw"></i> <i class="fa fa-caret-
down"></i>
        </a>
        <ul class="dropdown-menu dropdown-user">
            <!--<li><a href="#"><i class="fa fa-user fa-fw"></i> Profil
Pengguna</a>
            </li>
            <li class="divider"></li>-->
            <li><a href=" ../auth/logout.php"><i class="fa fa-sign-out fa-
fw"></i> Keluar</a>
            </li>
        </ul>
    <!-- /.dropdown-user -->
</li>
<!-- /.dropdown -->
</ul>

```

```

<!-- /.navbar-top-links -->
<div class="navbar-default sidebar" role="navigation">
  <div class="sidebar-nav navbar-collapse">
    <ul class="nav" id="side-menu">
      <?php
        if($_SESSION['hak'] == "PPIC" || $_SESSION['hak'] ==
"admin" || $_SESSION['hak'] == "gudang" || $_SESSION['hak'] ==
"produksi"){
          ?>
          <li>
            <a href="<?=base_url('dashboard/index.php')?>"><i
class="fa fa-dashboard fa-fw"></i> Dashboard</a>
          </li>
          <?php
            }
          ?>
          <?php
            if($_SESSION['hak'] == "admin" || $_SESSION['hak'] ==
"PPIC"){
              ?>
              <li>
                <a href="#"><i class="glyphicon glyphicon-book fa-
fw"></i> Data Master<span class="fa arrow"></span></a>
                <ul class="nav nav-second-level">
                  <li>
                    <a href="<?=base_url('bahanbaku/data.php')?>"><i
class="fa fa-list-alt fa-fw"></i> Data Bahan Baku</a>
                  </li>
                  <li>
                    <a href="<?=base_url('sparepart/data.php')?>"><i
class="fa fa-list-alt fa-fw"></i> Data Spare Part</a>

```

```
</li>
<li>
  <a href="<?=base_url('customer/data.php')?>"><i
class="fa fa-list-alt fa-fw"></i> Data Customer</a>
</li>
<li>
</li>
</ul>
</li>
<?php
}
?>
<?php
if($_SESSION['hak'] == "admin" || $_SESSION['hak'] ==
"PPIC"){
?>
<li>
  <a href="<?=base_url('spk/data.php')?>"><i class="fa fa-
edit fa-fw"></i> Perintah Kerja</a>
</li>
<?php
}
?>
<?php
if($_SESSION['hak'] == "produksi"){
?>
<li>
  <a href="<?=base_url('valspk/data.php')?>"><i class="fa fa-
edit fa-fw"></i> Perintah Kerja</a>
</li>
<?php
```

```
    }
?>
<?php
    if($_SESSION['hak'] == "admin" || $_SESSION['hak'] ==
"produksi"){
    ?>
    <li>
    <li>
        <a href="<?=base_url('permintaan/data.php')?>"><i
class="fa fa-edit fa-fw"></i> Permintaan Bahan Baku</a>
    </li>
    <?php
        }
    ?>
    <?php
    if($_SESSION['hak'] == "admin" || $_SESSION['hak'] ==
"gudang"){
    ?>
    <li>
        <a href="<?=base_url('pengiriman/data.php')?>"><i
class="fa fa-edit fa-fw"></i> Pengiriman Bahan Baku</a>
    </li>
    <?php
    if($_SESSION['hak'] == "produksi"){
    ?>
    <li>
        <a href="<?=base_url('valpengiriman/data.php')?>"><i
class="fa fa-edit fa-fw">
        </i> Pengiriman Bahan Baku</a>
    </li>
    <?php
```

```

    }
?>
<li>
    <a href="<?=base_url('pemotongan/data.php')?>"><i
class="fa fa-edit fa-fw"></i> Pemotongan Bahan Baku Mentah</a>
</li>
<?php
    }
?>
<?php
    if($_SESSION['hak'] == "admin" || $_SESSION['hak'] ==
"gudang" || $_SESSION['hak'] == "PPIC"){
?>
<li>
    <a href="#"><i class="glyphicon glyphicon-book fa-
fw"></i> Laporan<span class="fa arrow"></span></a>
    <ul class="nav nav-second-level">
        <!--<li>
            <a href="<?=base_url('pemakaian/data.php')?>"><i
class="fa fa-file-text fa-fw"></i> Laporan Pemakaian Bahan Baku</a>
        </li-->
        <li>
            <a href="<?=base_url('ketersediaan/data.php')?>"><i
class="fa fa-file-text fa-fw"></i> Laporan Ketersediaan Bahan Baku</a>
        </li>
        <li>
            <a href="<?=base_url('pkbb/data.php')?>"><i class="fa
fa-file-text fa-fw"></i> Laporan Rekapitulasi Perintah Kerja</a>
        </li>
        <li>
        </li>
    </ul>
</li>

```

```

        </ul>
    </li>
    <?php
        }
    ?>
</ul>
</div>
</div>
</nav>
<div id="page-wrapper">
    <div class="container-fluid">

```

11. Footer

```

    </div>
</div>
</div>
<!-- /#page-wrapper -->
</div>
<!-- /#wrapper -->
    <!-- Bootstrap Core JavaScript -->
    <script src="../../vendor/bootstrap/js/bootstrap.min.js"></script>
    <!-- Metis Menu Plugin JavaScript -->
    <script src="../../vendor/metisMenu/metisMenu.min.js"></script>
    <!-- Morris Charts JavaScript -->
    <script src="../../vendor/raphael/raphael.min.js"></script>
    <script src="../../vendor/morrisjs/morris.min.js"></script>
    <script src="../../data/morris-data.js"></script>
    <!-- Custom Theme JavaScript -->
    <script src="../../dist/js/sb-admin-2.js"></script>
</body>
</html>

```


14. Form Bon Permintaan Material

PT. SUKSES CIPTA MAKMUR
 Tanggal: 04/08/2023
 Halaman: 1/1

BON PERMINTAAN MATERIAL

No. Bon: 0000000001

No	No. SPK	Nama & Spesifikasi Material	Jumlah Total	Jumlah per hari	Jumlah Hari	Keterangan
1	0000000001	BESI	100	10	10	
2	0000000001	STAINLESS	100	10	10	

Disetujui: 
 Tanggal: 04/08/2023

15. Kartu Stok Bahan Baku

A.B.C.D. INDUSTRIAL MANUFACTURING

Formulir No. 001
 Tanggal: 04/08/2023

No	No. SPK	Nama	Spesifikasi	Saldo Awal	Saldo Akhir	Perubahan
1	001	Besi	100	100	100	0
2	002	Stainless	100	100	100	0
3	003	Alumunium	100	100	100	0
4	004	Stainless	100	100	100	0
5	005	Alumunium	100	100	100	0
6	006	Stainless	100	100	100	0
7	007	Alumunium	100	100	100	0
8	008	Stainless	100	100	100	0
9	009	Alumunium	100	100	100	0
10	010	Stainless	100	100	100	0

16. Form Pengiriman Material ke Produksi

FORMA PENGIRIMAN MATERIAL KE PRODUKSI

Tanggal: 04/08/2023

No	No. SPK	Nama	Spesifikasi	Jumlah	Tanggal	Waktu
1	001	Besi	100	100	04/08/2023	08.00
2	002	Stainless	100	100	04/08/2023	08.00
3	003	Alumunium	100	100	04/08/2023	08.00
4	004	Stainless	100	100	04/08/2023	08.00
5	005	Alumunium	100	100	04/08/2023	08.00
6	006	Stainless	100	100	04/08/2023	08.00
7	007	Alumunium	100	100	04/08/2023	08.00
8	008	Stainless	100	100	04/08/2023	08.00
9	009	Alumunium	100	100	04/08/2023	08.00
10	010	Stainless	100	100	04/08/2023	08.00

17. Form Laporan Harian Kerja Cutting

PT. GUNUNG CITRA MANUNG CIRIK, SUKSES		LAPORAN HARIAN KERJA CUTTING		NO. DAFTAR	
NO	ALOKASI	ALOKASI	ALOKASI	ALOKASI	ALOKASI
1	1000	1000	1000	1000	1000
2	1000	1000	1000	1000	1000
3	1000	1000	1000	1000	1000
4	1000	1000	1000	1000	1000
5	1000	1000	1000	1000	1000
6	1000	1000	1000	1000	1000
7	1000	1000	1000	1000	1000
8	1000	1000	1000	1000	1000
9	1000	1000	1000	1000	1000
10	1000	1000	1000	1000	1000
11	1000	1000	1000	1000	1000
12	1000	1000	1000	1000	1000
13	1000	1000	1000	1000	1000
14	1000	1000	1000	1000	1000
15	1000	1000	1000	1000	1000
16	1000	1000	1000	1000	1000
17	1000	1000	1000	1000	1000
18	1000	1000	1000	1000	1000
19	1000	1000	1000	1000	1000
20	1000	1000	1000	1000	1000
21	1000	1000	1000	1000	1000
22	1000	1000	1000	1000	1000
23	1000	1000	1000	1000	1000
24	1000	1000	1000	1000	1000
25	1000	1000	1000	1000	1000
26	1000	1000	1000	1000	1000
27	1000	1000	1000	1000	1000
28	1000	1000	1000	1000	1000
29	1000	1000	1000	1000	1000
30	1000	1000	1000	1000	1000

18. Rekapitulasi Surat Perintah Kerja

NO	ALOKASI	ALOKASI	ALOKASI	ALOKASI	ALOKASI
1	1000	1000	1000	1000	1000
2	1000	1000	1000	1000	1000
3	1000	1000	1000	1000	1000
4	1000	1000	1000	1000	1000
5	1000	1000	1000	1000	1000
6	1000	1000	1000	1000	1000
7	1000	1000	1000	1000	1000
8	1000	1000	1000	1000	1000
9	1000	1000	1000	1000	1000
10	1000	1000	1000	1000	1000
11	1000	1000	1000	1000	1000
12	1000	1000	1000	1000	1000
13	1000	1000	1000	1000	1000
14	1000	1000	1000	1000	1000
15	1000	1000	1000	1000	1000
16	1000	1000	1000	1000	1000
17	1000	1000	1000	1000	1000
18	1000	1000	1000	1000	1000
19	1000	1000	1000	1000	1000
20	1000	1000	1000	1000	1000
21	1000	1000	1000	1000	1000
22	1000	1000	1000	1000	1000
23	1000	1000	1000	1000	1000
24	1000	1000	1000	1000	1000
25	1000	1000	1000	1000	1000
26	1000	1000	1000	1000	1000
27	1000	1000	1000	1000	1000
28	1000	1000	1000	1000	1000
29	1000	1000	1000	1000	1000
30	1000	1000	1000	1000	1000

LAMPIRAN D

WAWANCARA

19. Proses Bisnis

- Saya : Mas Sagus, saya mau tanya tugas mas di SCM ini apa?
- Mas Sagus : Saya bertugas mengelola surat perintah kerja (SPK), dari Bu Ambar memberikan SPK lalu saya yang mengelola sampai saya mengirimkan bahan baku ke produksi.
- Saya : Kalau saya boleh tahu bagaimana alur proses dari tugas yang mas kerjakan disini seperti apa?
- Mas Sagus : Ya, pertama Bu Ambar memberikan SPK dari hasil PO *customer* lalu SPK masuk ke QC untuk menentukan detail gambar, kemudian QC memberikan ke produksi SPK dan detail gambar, produksi membuat bon *material* yang kemudian diberikan ke saya. Lalu saya memberitahu ke gudang ada SPK baru kemudian gudang mengecek ketersediaan bahan baku, jika mencukupi bahan baku langsung dikirim kalau tidak mencukupi bahan baku dipotong lebih dahulu.
- Saya : Berarti SPK yang dibuat oleh Bu Ambar punya berapa rangkap untuk diberikan ke QC dan produksi mas?
- Mas Sagus : SPK ada 5 rangkap, 2 rangkap untuk PPIC, 1 rangkap untuk QC dan 2 rangkap untuk produksi.
- Saya : Kemudian detail gambar itu fungsinya apa terhadap SPK?
- Mas Sagus : Detail gambar itu untuk mengetahui ukuran part yang ingin dibuat beserta bahan bakunya.
- Saya : Maksudnya mas?
- Mas Sagus : Jadi di SPK ada nama part dan *material* nah detail gambar berisi ukuran-ukuran yang dibutuhkan untuk membuat part tersebut.
- Saya : Kalau part ini selalu bahan bakunya sama mas?

Mas Sagus : Tidak. Kadang 1 part bisa memakai beberapa bahan baku tapi tergantung SPK yang dibuat kan disitu ada nama *material* yang dibutuhkan.

Saya : Selanjutnya mas, apa lagi prosesnya?

Mas Sagus : Lebih lanjutnya kita turun ke gudang saja.

20. Penjelasan lebih lanjut perihal SPK di gudang

Mas Sagus : ini rekapitulasi SPK.

Saya : Rekapitulasi SPK ini diisi oleh siapa mas?

Mas Sagus : Diisi sama Jarot/Agus tergantung *shift* mereka.

Saya : Oh, lalu Rekapitulasi ini dilakukan berapa kali mas?

Mas Sagus : Dilakukan 1 bulan sekali.

Saya : Mas ini siapa yang melakukan pemotongan bahan baku mentah?

Mas Sagus : Jejen, Oji sama Kosim.

Saya : Kalau yang menerima bahan baku dari sini ke produksi siapa?

Mas Sagus : Joko sama Sainan.

21. Penjelasan lebih lanjut perihal SPK dengan bagian yang terkait

Saya : Mas Jejen, mau nanya. Tugas mas di gudang ini apa?

Mas Jejen : Saya tugasnya melakukan pemotongan bahan baku mentah.

Saya : Kalau proses pemotongan bahan baku mentah seperti apa mas?

Mas Jejen : Prosesnya disini kalau ada bahan baku yang sudah dipotong kurang dari minimum stok baru ada pemotongan bahan baku, lanjut nulis di laporan harian kerja *cutting* dan kartu stok.

Saya : Oh berapa minimum stoknya mas?

Mas Jejen : *Simplenya* aja ya stok yang kurang dari 100 pcs baru dilakukan pemotongan.

Saya : Untuk pemotongan ini jumlah tiap bahan baku yang dipakai apakah sama mas?

Mas Jejen : Tidak, setiap bahan baku kan diameternya berbeda jadi jumlah untuk pemotongan tiap mesin pasti berbeda-beda.

Saya : Oke deh mas. Terima kasih infonya.

Saya : Mas Jarot, mau nanya. Tugas mas di gudang ini apa?

Mas Jarot : Saya tugasnya melakukan pengiriman bahan baku ke produksi sama rekapitulasi SPK.

Saya : Seperti apa mas alur prosesnya?

Mas Jarot : Prosesnya disini kan dari bon *material*, produksi minta bahan baku sesuai jumlah yang diminta. Kemudian saya catat di *form* pengiriman *material* ke produksi. Habis itu saya nulis di kartu stok bahan baku yang keluar bererapa sesuai dengan nama bahan baku, lanjut nulis di rekapitulasi SPK sesuai nomer SPK dan nama bahan baku.

Saya : Sebentar mas, berarti jumlah yg diminta produksi pada bon *material* sama yang di kirim selalu sama?

Mas Jarot : Ya selalu sama jumlahnya.

Saya : Kalau untuk rekapitulasi SPK ini untuk mengetahui selesainya SPK dari mana mas?

Mas Jarot : Kan ini ada sisa, kalau sisahnya sudah 0 berarti SPK tersebut sudah selesai.

Saya : Kalau misalnya ada SPK yang belum selesai di bulan ini bagaimana mas?

Mas Jarot : SPK yg belum selesai akan ditulis lagi pada rekapitulasi SPK bulan depan.

Saya : Oh seperti itu ya mas. Terima kasih infonya.

22. Mempertanyakan permasalahan yang di rasakan oleh karyawan PPIC

Saya : Mas Sagus saya mau tanya, kesulitan mas untuk mengelola SPK itu apa ya?

Mas Sagus : Kesulitannya itu mengetahui selesainya perintah kerja untuk Bu Ambar karna saya lebih sering berada di gudang. Sama rekapitulasi SPK masih ditulis tangan kadang-kadang tulisannya

tidak terbaca mungkin karena buru-buru nulisnya. Itu sih permasalahannya.

Saya : Seperti itu ya permasalahannya mas. Terima kasih banyak atas informasinya ya mas.

23. Menunjukkan aplikasi yang dibuat (foto)

