BAB II
TINJAUAN PUSTAKA

Pada bab tinjauan pustaka akan diuraikan berbagai macam teori dan referensi yang akan menunjang dalam penelitian ini. Teori dan referensi tersebut berkaitan dengan sistem informasi, pengembangan *database*, perawatan mesin dan sebagainya.

2.1 Penelitian Terdahulu

Ada beberapa penelitian sebelumnya yang berhubungan dengan penelitian perancangan sistem informasi penjadwalan *preventive mintenance*.

1. Suriono (2013), penelitian ini dilakukan di PT. XYZ. Permasalahan utama yang terjadi adalah kegiatan *maintenance* (pemeliharaan) yang belum diterapkan dengan baik karena padatnya jadwal produksi, sehingga seringkali menyebabkan kegiatan penggantian komponen baru dilakukan ketika terjadi kerusakan pada mesin secara mendadak. Analisis dilakukan dengan menentukan mesin dan komponen kritis, dan kemudian menghitung nilai *Mean Time To Failure* (MTTF) dan *Mean Time To Repair* (MTTR) untuk mendapatkan nilai *age replacement*, *availability*, *reliability* dan *downtime*. Hasilnya adalah peningkatan *realibility* dan *availibility* serta penurunan *downtime* melalui penerapan *preventive maintenance.*
2. William (2013), melakukan penelitian di PT. HB. Permasalahan yang dihadapi adalah frekuensi kerusakan yang cukup banyak dan waktu perbaikan cukup lama. Tujuan penelitian ini ialah mengembangkan sistem informasi penjaadwalan perawatan prediktip yang dikembangkan dengan metode *system development life cycle* dan rekayasa ulang proses bisnis perawatan.
3. Melladya (2014), melakukan penelitian di PT. Adi Putro Wirasejati. Permasalahan yang terjadi adalah seringnya keterlambatan perawatan mesin karena tidak mempunyai sistem penyajian informasi yang baik dan sistem masih dilakukan secara manual. Selain itu perusahaan juga tidak memiliki pengelolaan *inventory sparepart* yang baik sehingga sering kali pencarian sparepart yang dibutuhkan memakan waktu lama. Untuk itu peneliti merancang aplikasi penjadwalan perawatan mesin disertai dengan pengelolaan *invetory* database *sparepart* mesin.

Tabel 2.1 Perbandingan Penelitian Pendahuluan

Peneliti	Objek	Tools	Metode	Hasil Penelitian
Suriono (2013)	Mesin Die Casting PT. XYZ	PHP, MySQL dan XAMM	Age Replacement	Software untuk melakukan penjadwalan perawatan mesin, sehingga menghasilkan peningkatan realibility dari mesin dan komponen kritis terpilih.
William (2013)	Mesin produksi di PT. HB	Delphi	SDLC (System Development Life Cycle)	Dari 3 skenario waktu penggantian yang diuji cobakan, didapakan waktu optimal untuk penggantian komponen mesin B. Flute.
Melladya (2014)	Mesin – mesin di PT. Adiputro Wirasejati	Ms. Acces 2013 dan VBA	*Software Prototyping*	*Prototype* sistem informasi manajemen perawatan mesin disertai pengelolaan database sparepart mesin yang dikelompokkan berdasarkan *Group Technology*.
Herlangga (2015)	Mesin Pounding di PT. XYZ	Ms. Acces 2013 dan VBA	*Software Prototyping*	Perancangan *prototype* sistem informasi manajemen yang bisa menjadwalkan preventive maintenance secara otomatis berdasarkan usia pemakaian sparepart beserta pengelolaan database sparepart mesin pounding.

2.2 Sistem *Database*

Database merupakan kumpulan data yang tersimpan dalam suatu *hardware* dan saling terintegerasi satu sama lain yang membentuk suatu informasi yang berguna. *Database* diatur sedemikian rupa agar mudah dalam mengolah maupun mencari data yang dibutuhkan.

2.2.1 Komponen Sistem *Database*

Di dalam suatu sistem *database* terdapat empat komponen penting yaitu:

1. Data

   Merupakan dasar sumber daya organisasi yang diperlukan untuk memproses informasi. Data dapat berbentuk teks, gambar, audio maupun video. Sumber daya informasi umumnya diatur, disimpan dan diakses oleh berbagai pengelolaan sumber daya data ke dalam *database* dan dasar pengetahuan (Lajamudin, 2005). Ditinjau dari jumlah pemakai, data dapat dikelompokan menjadi 2 yaitu:

   a. *Data single user*, yaitu suatu sistem *database* yang hanya dioperasikan oleh satu pemakai.
   b. *Data multi user*, yaitu satu atau lebih pemakai bisa beroperasi secara bersamaan ke dalam *database*, sehingga data yang ada dalam *database* dapat digunakan bersama dan saling terintegrasi.

2. *Hardware*

   Hardware adalah semua peralatan yang digunakan dalam memproses informasi, misalnya komputer dan periferalnya, lembar kertas, *disk magnetic* atau optik dan *flash disk* (Mulyanto, 2009).

3. *Software*

   Software merupakan sekumpulan perintah/fungsi yang ditulis dengan aturan tertentu untuk memerintahkan komputer agar melaksanakan sesuatu (Ladjamudin, 2005). Fungsi *software* dalam hal ini sebagai penghubung antara *database* dengan pemakai yang disebut sebagai *Database Management System* (DBMS) atau *database* manajer. Tugas dari DBMS ini adalah melakukan semua kebutuhkan akses data.

4. *User*

   User atau pemakai *database* dapat dikelompokan menjadi tiga yaitu:

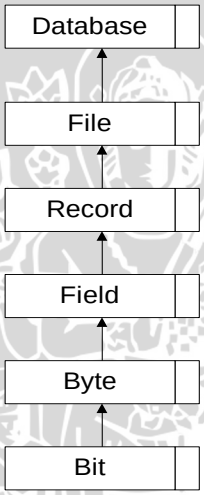
   a. *Programmer* aplikasi, yaitu orang yang bertanggung jawab untuk menulis program aplikasi.
   b. *End user*, yaitu orang yang menggunakan data dalam *database* untuk kebutuhan tugas atau fungsinya.
   c. Administrator *database*, yaitu orang yang bertanggung jawab pada keseluruhan sistem *database*.

2.2.2 Pengertian Data

Menurut kamus besar bahasa Indonesia, data diterjemahkan sebagai istilah yang berasal dari kata "*datum*" yang berarti fakta atau bahan-bahan keterangan kumpulan kejadian yang diangkat dari suatu kenyataan. Data dapat berupa angka-angka, huruf, atau simbol-simbol khusus atau gabungan darinya. Data mentah masih belum bisa bercerita banyak, sehingga perlu diolah lebih lanjut. Pengolahan data adalah manipulasi dari data ke dalam bentuk yang lebih berguna dan lebih berarti berupa suatu informasi (McLeod, 2004:167).

2.2.3 Hierarki Data

Berikut merupakan komponen penyusun database yang dimulai dari satuan terkecil dari satuan terkecil sebuah data.



Gambar 2.1 Hierarki Data
Sumber : Kusrini (2007)

Berdasarkan Gambar 2.1, pengorganisasian data dapat dibagi menjadi enam tingkatan, yaitu:

1. *Bit* adalah suatu sistem angka biner yang terdiri atas dua macam nilai saja, yaitu 0 dan 1. Sistem angka biner merupakan dasar-dasar yang dapat digunakan untuk komunikas antara manusia dan mesin (komputer) yang merupakan sekumpulan komponen elektronik dan hanya dapat membedakan dua keadaan saja (*on* dan *off*). Jadi bit adalah unit terkecil pembentuk data.
2. *Byte* adalah bagian terkecil yang dapat dialamatkan dalam memori. *Byte* merupakan sekumpulan *bit* yang secara konvensional terdiri atas delapan *bit*. Satu *byte* digunakan untuk mengkodekan satu buah karakter dalam memori. Contoh: Kode ASCII untuk J ialah 10101010. Jadi byte adalah kumpulan *bit* yang membentuk suatu karakter (huruf,

angka atau tanda). Dengan kombinasi 8 *bit*, dapat diperoleh 256 karakter (= 2 dipangkat 8).

3. *Field* atau kolom adalah unit terkecil yang disebut data. *Field* merupakan sekumpulan *byte* yang mempunyai makna. Contoh: Joni yang merupakan *field* nama. Jadi *field* ibarat kumpulan karakter yang membentuk suatu kata.
4. *Record* atau baris adalah kumpulan item yang secara *logic* saling berhubungan. Setiap *record* dapat dikenali oleh sesuatau yang mengenalinya, yaitu *field* kunci. Jadi *record* ibarat kumpulan data yang membentuk satu kalimat yang berarti.
5. *File* atau tabel adalah kumpulan *record* yang sejenis dan secara logis berhubungan. Pembuatan dan pemeliharaan *file* adalah faktor yang sangat penting dalam sistem informasi manajemen yang memakai komputer. Jadi tabel ibarat kumpulan baris/*record* yang membentuk satu tabel.
6. *Database* merupakan kumpulan *file-file* yang berhubungan secara logis dan digunakan secara rutin pada operasi-operasi sistem informasi manajemen. Semua *database* umumnya berisi elemen-elemen data yang disusun ke dalam *file-file* yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, tersimpan di *hardware* komputer dengan *software* untuk melakukan manipulasi data untuk kegunaan tertentu. Jadi, suatu *database* adalah menunjukkan suatu kumpulan tabel yang dipakai dalam suatu lingkup perusahaan atau instansi untuk tujuan tertentu.

2.2.4 Tujuan Sistem *Database*

Tujuan utama dari sistem *database* adalah praktis dan efisien. Secara umum, tujuan dari sistem *database* adalah sebagai berikut:

1. Isolasi data, menempatkan tiap data pada tempatnya masing-masing.
2. Multi *user*, saat perusahaan mengambil pendekatan berorientasi masalah, pertama didefinisikan kemudian pengambilan keputusan untuk penyelesaian masalah tersebut dan untuk pengambilan keputusan diperlukan informasi.

Sedangkan tujuan dari manajemen sistem *database* adalah sebagai berikut:

1. Menyediakan tempat penyimpanan massal untuk data yang relevan.
2. Memudahkan pemakai dalam mengakses data.
3. Memungkinkan respon yang segera atas permintaan data dari pemakai.
4. Melakukan modifikasi terakhir dengan segera pada *database*.
5. Memungkinkan perkembangan lebih lanjut dalam sistem *database*.
6. Meminimasi duplikasi dan redudansi dalam penyimpanan data.

7. Memungkinkan digunakan secara serentak dan bersamaan
8. Melindungi data dari gangguan kerusakan atau pemakaian oleh orang yang tidak terotorisasi.

2.2.5 *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram (ERD) atau diagram hubungan entitas merupakan model data berupa notasi grafis dalam pemodelan data konseptual pada basis data yang menggambarkan hubungan antara penyimpan atau objek dasar data. Data model merupakan representasi abstrak dari data tentang entitas, kejadian, aktifitas, dan asisiasinya dalam suatu organisasi. Tujuan dari pemodelan data adalah untuk mempermudah perancangan dan pengaksesan database.

Menurut Whitehorn dan Marklyn (2003) *database* relasional adalah suatu database yang menggunakan model relasional, dimana model tersebut menyediakan mekanisme-mekanisme yang kita perlukan untuk menyimpan dan memanipulasi data yang kompleks dengan cara yang sedemikian rupa sehingga kita dapat meng-query data itu dan memperoleh jawaban-jawaban yang kita perlukan. Ada beberapa komponen pokok dalam database *relasional* menurut Thomas (2001) yaitu :

1. *Entity*, adalah objek-objek terpisah (orang, tempat, benda, konsep, atau peristiwa) dalam suatu organisasi yang akan direpresentasikan ke dalam *database (*menjadi tabel).
2. *Attribute*, adalah properti yang menggambarkan aspek-aspek yang dimiliki suatu objek yang akan dicatat (menjadi kolom).
3. *Relationship* (relasi), adalah hubungan-hubungan yang dapat terjadi antara dua *entity*.
4. Data disimpan dalam suatu tabel berbentuk dua dimensi, field (kolom) dan record (baris).

2.2.6 Normalisasi

Perancangan *database* yang kompak dan efisien dalam penggunaan ruang penyimpanan, kecepatan pengambilan data dan mudah dalam memanipulasi data sangat penting. Untuk tercapainya tujuan di atas kita membutuhkan struktur data yang sederhana dan stabil. Proses untuk mengubah suatu bentuk struktur data menjadi lebih sederhana dan stabil disebut dengan normalisasi. Tujuan normalisasi menurut Whitehorn dan Marklyn (2004) adalah sebagai berikut.

1. Untuk mencari dan mengelompokkan seluruh sifat yang berkenaan dengan objek tertentu.
2. Untuk menghilangkan informasi yang berlebihan.
3. Untuk memberikan identifikasi unik bagi *record* individu (*primary key*).

Proses dalam menganalisa data dalam normalisasi dilakukan dengan beberapa tahapan sehingga terbentuk *normal form*. Untuk menciptakan relasi *normal form*, dapat dibentuk dengan mengaplikasikan suatu aturan sederhana tanpa mengurangi fungsi ketergantungan (relasi antara atribut) dalam relasi. Berikut ini diberikan gambaran tentang aturan tersebut secara ringkas.

1. *First normal form* (1NF).

   Hapuskan semua elemen yang berulang, sehingga hanya terdapat satu nilai untuk setiap perpotongan baris dan kolom pada tabel.
2. *Second normal form* (2NF)

   Setiap atribut bukan kunci (*non key*) yang terdapat pada relasi harus tergantung pada *primary key*. Jika hal ini tidak erpenuhi maka perlu dibuat tabel baru.
3. *Third normal form* (3NF)

   Jika terdapat kesaling tergantungan antar dua atribut yang bukan merupakan atribut kunci, maka harus dibuat tabel baru. Dengan kata lain setiap relasi (tabel) hanya memuat satu kepentingan.

2.3 Perawatan

Definisi perawatan (*maintenance*) menurut Patrick (2001) adalah suatu kegiatan untuk memelihara dan menjaga fasilitas yang ada serta memperbaiki, melakukan penyesuaian atau penggantian yang diperlukan untuk mendapatkan suatu kondisi operasi produksi agar sesuai dengan perencanaan yang ada.

Tujuan utama dilakukannya perawatan menurut Patrick (2001) yaitu:

1. Mempertahankan kemampuan alat atau fasilitas produksi guna memenuhi kebutuhan yang sesuai dengan target serta rencana produksi.
2. Mengurangi pemakaian dan penyimpangan diluar batas dan menjaga modal yang diinvestasikan dalam perusahaan selama jangka waktu yang ditentukan sesuai dengan kebijaksanaan perusahaan.
3. Menjaga agar kualitas produk berada pada tingkat yang diharapkan guna memenuhi apa yang dibutuhkan produk itu sendiri dan menjaga agar kegiatan produksi tidak mengalami gangguan.

4. Memperhatikan dan menghindari kegiatan – kegiatan operasi mesin serta peralatan yang dapat membahayakan keselamatan kerja.
5. Mencapai tingkat biaya serendah mungkin, dengan melaksanakan kegiatan *maintenance* secara efektif dan efisien untuk keseluruhannya.
6. Mengadakan suatu kerjasama yang erat dengan fungsi – fungsi utama lainnya dari suatu perusahaan, dalam rangka untuk mencapai tujuan utama perusahaan yaitu tingkat keuntungan atau *return investment* yang sebaik mungkin dan total biaya serendah mungkin.

2.3.1 Jenis Perawatan

Perawatan berdasarkan jenisnya dapat dibedakan menjadi tiga yaitu *preventive maintenance, corrective maintenance* dan *predictive maintenance,* yang bisa dijelaskan sebagai berikut.

1. *Preventive maintenance* (PM)

Menurut pendapat Patrick (2001) PM adalah kegiatan pemeliharaan dan perawatan yang dilakukan untuk mencegah timbulnya kerusakan-kerusakan yang tidak terduga dan menemukan kondisi atau keadaan yang dapat menyebabkan fasilitas produksi mengalami kerusakan pada waktu proses produksi. Kegiatan seperti ini bisa meliputi pelumasan, penyetingan ulang, penggantian sparepart dan sebagainya. Pada intinya kegiatan seperti ini ditujukan agar mesin produksi tidak sampai mengalami kerusakan sparepart yang berkemungkinan besar bisa menyebabkan kerusakan *sparepart* lainnya.

2. *Corrective maintenance* (CM)

Menurut pendapat Patrick (2001) CM merupakan kegiatan perawatan yang dilakukan setelah mesin atau fasilitas produksi mengalami kerusakan atau gangguan sehingga tidak dapat berfungsi dengan baik. Tindakan seperti ini sering disebut dengan perbaikan. Biasanya perbaikan ini dilakukan karena ada sparepart yang rusak atau mengalami kegagalan.

Kerusakan yang terjadi tidak dapat dihindari karena memang berada diluar perhitungan. Hal seperti ini terjadi apabila PM tidak dijalankan dengan baik, misalnya penundaan jadwal PM atau memang tidak diberlakukan PM sama sekali dengan alasan mengejar target produksi.

3. Predictive Maintenance

Predictive maintenance merupakan bagian dari usaha pencegahan terjadinya kerusakan dimana pelaksanaannya didasarkan pada kondisi mesin itu sendiri. Untuk itu

dalam melakukan strategi perawatan diperlukan pemeriksaan mesin secara rutin untuk mendeteksi gejala kerusakan mesin. *Predictive maintenance* disebut juga sebagai *condition based maintenance* atau *machinery condition monitoring* yang artinya penentuan kondisi mesin dengan cara memeriksa mesin secara rutin sehingga keandalan mesin dan keselamatan kerja dapat dijamin. Menurut Mobley(2002), *predictive maintenance* bertujuan meningkatkan produktivitas, kualitas dan keseluruhan efektivitas dari proses manufaktur. Sebenarnya *predictive maintenance* adalah *preventive maintenance* yang didasarkan oleh kondisi sesungguhnya dari mesin. Metode pelaksanaan pemeliharaan *predictive maintenance*:

a. Pemilihan peralatan

   Tidak perlu seluruh peralatan mesin dipelihara secara *predictive maintenance*, tetapi memilih peralatan-perlatan yang kritis atau mahal, juga dipengaruhi oleh fungsi dan kondisi spesifik suatu peralatan.

b. Pengumpulan Data Sejarah Mesin

   Riwayat mesin dapat dipakai sebagai pendekatan teknik pemantauan dan analisa pemeliharaan. Data/ informasi tersebut dapat berupa: data desain, data sejarah mesin dan data sejarah operasi mesin lain yang sejenis.

c. Pemasangan alat-alat sensor

   Pemasangan alat sensor pada bagian-bagian tertentu untuk dapat memantau kondisi peralatan sangat diperlukan. Pemantauan itu meliputi: vibrasi, temperatur, tekanan, laju aliran, korosi dan lain sebagainya.

d. Pemantauan rutin

   Pemantauan dilaksanakan ketika unit sedang beroperasi atau unit sedang stop, tergantung pada objek yang akan dipantau.

Keuntungan yang bisa diperoleh jikan perusahaan menerapkan *predictive maintenance* adalah :

a. Prediksi kondisi komponen mesin dapat diketahui lebih dini, sehingga pemeliharaan lebih efektif.
b. Umur asset/ material dapat diketahi dari prediksi kondisi material sehingga persiapan pemeliharaan, perencanaan dan pengadaan material terencana dengan baik.
c. Dengan *predictive maintenance*, produktivitas mesin lebih optimal karena tidak adanya *downtime* akibat gangguan yang berarti.
d. Efisiensi, unjuk kerja dan pelayanan lebih optimal.

e. Keselamatan pengguna pembangkit lebih terjamin

Sedangkan kerugian yang diperoleh dari penerapan *predictive maintenance* adalah sebagai berikut:

Pada hakekatnya kerugian dari sisi pelaksanaan tidak ada, hanya memerlukan biaya tambahan, contoh untuk pengadaan peralatan monitor dan sensor-sensor. Tetapi biaya yang dikeluarkan akan sebanding dengan hasil yang didapatkan, karena bisa saja jika perusahaan tidak menerapkan *predictive maintenance*, sehingga mesin sampai mengalami kerusakan, biaya yang dikeluarkan mungkin akan lebih besar dari biaya penerapan *predictive maintenance*.

2.4 *Mean Time To Failure* (MTTF)

Menurut Kostas (1981:73), *Mean Time Between Failure* (MTTF) adalah rata-rata interval waktu kerusakan yang terjadi saat mesin selesai diperbaiki sampai mesin tersebut mengalami kerusakan kembali. MTTF dapat dihitung dengan persamaan sebagai berikut.

$$\textit{Mean Time to Failure} = \frac{Total\ Time\ to\ Failure}{Number\ of\ Failure} \quad (2\text{-}1)$$

2.5 *Mean Time To Repair* (MTTR)

Mean Time to Repair (MTTR) merupakan waktu rata-rata dari interval waktu untuk melakukan perbaikan yang dibutuhkan oleh suatu komponen atau sistem. Menurut Kostas (1981:73), MTTR diperoleh dengan rumus sebagai berikut.

$$\textit{Mean Time to Repair} = \frac{Total\ Repair\ Time}{Number\ of\ Repair} \quad (2\text{-}2)$$

2.6 *Software Prototyping*

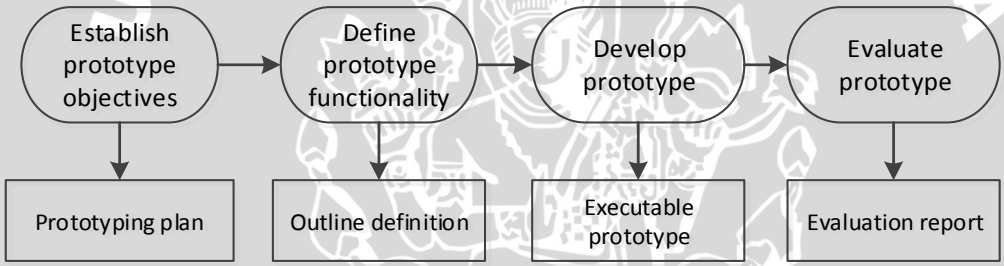
Menurut Kendall (2003), *prototyping* adalah suatu teknik pengumpulan data yang sangat berguna melengkapi siklus hidup pengembangan sistem tradisional. Saat penganalisis sistem menggunakan *prototyping*, mereka berusaha mencari reaksi, saran – saran, inovasi, rencana revisi pengguna untuk membuat peningkatan terhadap *protoype* sekaligus memodifikasi rencana sistem dengan biaya dan gangguan maksimum.

Istilah *prototyping* memiliki beberapa arti berbeda, dimana empat diantaranya paling banyak digunakan. Definisi *protoype* pertama adalah penyusunan *protoype patch*. Definisi kedua adalah *protoype* operasional yang digunakan untuk menguji fitur – fitur desain tertentu. Konsepsi ketiga adalah menciptakan *protoype* pertama yang benar – benar operasional. Sedangkan jenis *protoype* yang keempat adalah *protoype* fitur – fitur terpilih

yang memiliki beberapa fitur – fitur sistem inti. Empat petunjuk utama untuk mengembangkan suatu *protoype* adalah:

1. Bekerja menurut modul – modul dapat dikendalikan
2. Membangun *protoype* dengan cepat
3. Memodifikasi *protoype*
4. Menekankan *interface* pengguna.

Sommerville (2011) menjelaskan bahwa terdapat empat tahapan dalam pegembangan *software* menggunakan *prototyping*. Langkah pertama yaitu menetapkan tujuan *protoype* dengan mengidentifikasi masalah pada sistem yang akan dibuat *prototype*-nya. Langkah kedua yaitu mendefinisikan fungsi *protoype* sehingga sesuai dengan kebutuhan *user*. Langkah ketiga yaitu mengembangkan *protoype* dengan merancang desain logis hingga implementasi konsep yang telah dibuat. Langkah terakhir yaitu evaluasi *protoype* dengan membandingkan hasil *protoype* dan kebutuhan *user*. Langkah-langkah pengembangan tersebut dapat dilihat pada Gambar 2.2.



Gambar 2.2 Model proses pengembangan *protoype*
Sumber: Sommerville (2011)

Berdasarkan Gambar 2.2 model proses pengembangan *prototype* yang lebih rinci dapat dijelaskan sebagai berikut:

1. Menetapkan tujuan *prototype*
   a. Mengidentifikasi masalah dalam sistem yang sedang berjalan dengan menggunakan analisis PIECES *(Performance-Information-Economy-Control-Eficiency-Services)*
   b. Menetapkan batasan-batasan (*constraints*) atau ruang lingkup *prototype* yang akan dirancang.
   c. Menetapkan tujuan dan manfaat dari *prototype*
2. Mendefinisikan fungsi *prototype*
   a. Membuat model kebutuhan sistem (*Requirement Modelling*)

      Suatu kebutuhan sistem dinyatakan dalam suatu *checklist* yang disebut SRC (*System Requirement Checklist*). SRC adalah fitur-fitur atau karakteristik yang

harus ada dalam sistem informasi untuk memenuhi kebutuhan bisnis (*business requirements*) dan yang dapat diterima pengguna. Model kebutuhan sistem ini digambarkan ke dalam lima kategori umum yaitu *output*, *input*, *process*, *performance*, dan *control*.

b. Membuat model data (*Data Modelling)*

Pada *data modelling*, sistem yang akan dirancang digambarkan dengan aliran data dan informasi yang dihasilkan dari dan untuk *entity* dalam sistem dengan *Data Flow Diagram* agar nampak jelas.

c. Membuat model proses (*Process Modelling*)

Menggambarkan logika atau aturan bisnis yang dapat dinyatakan dengan *flowchart* atau tabel aturan. Model proses nanti akan berguna dalam membangun algoritma program komputer.

d. Pengembangan Strategi (*Development Strategies*)

Development Strategies atau strategi pengembangan adalah merupakan tahap untuk menggambarkan kegiatan yang tersisa dalam tahap analisis sistem. Tahap ini menjelaskan transisi dari analisa sistem ke sistem desain, *prototyping*, dan pedoman desain sistem diakhiri dengan bagaimana pengembangan perangkat lunak yang dirancang.

3. Mengembangkan *prototype*

a. Langkah Desain

1) Desain *Database* Logis

Konsep model data yang telah terbentuk pada tahap *data modelling* dibawa ke bentuk yang logis. Desain logis terdiri dari daftar entitas untuk pembuatan tabel dan normalisasi tabel.

2) Desain *Database* Fisik

Desain fisik merupakan aktualisasi dari desain logis. Disini *entity* sudah berubah menjadi *table* dengan rancangan bentuk fisik.

3) Desain *User Interface* (UI)

Bertujuan untuk membuat rancangan dari tampilan sistem yang nantinya akan berinteraksi langsung dengan *user* (pengguna). Desain *user interface* harus siap digunakan dan sesuai dengan kebutuhan pengguna.

4) Desain Algoritma

Bertujuan untuk merancang tahapan proses apa saja yang harus dilakukan sehingga *input*, *user interface*, dan *database* menghasilkan *output* yang

diharapkan dan dapat ditampilkan, algoritma dapat dinyatakan dengan *flowchart* ataupun *pseudocode.*

b. Implementasi, langkah ini adalah membuat aplikasi pada tingkatan *prototype* dari spesifikasi dan konsep desain yang dirancang dengan melakukan pengembangan *database, module* dan *user interface*.

4. Mengevaluasi *prototype*

Pada langkah ini dilakukan pengujian terhadap program aplikasi yang telah dibuat. Pengujian program ini ditinjau dari tiga segi, yaitu verifikasi, validasi dan uji *prototype*.

a. Verifikasi, menguji apakah *prototype* berjalan sesuai yang telah direncanakan. Uji Verifikasi ini meliputi pengujian *hierarki menu, form, report* dan pengujian ketelitian.

b. Validasi, menguji apakah fungsi *prototype* yang dirancang telah merepresentasikan kebutuhan user yang meliputi lima kategori umum: *output*, *input*, proses, *performance*, dan *control* (SRC).

c. Uji *prototype* bertujuan untuk mengetahui apakah *prototype* dapat mengatasi masalah dan kelemahan sistem yang lama yang.

Menurut Wetherbe (1994), untuk mengidentifikasi dan mengevaluasi permasalahan menggunakan *software protoyping,* dapat menggunakan PIECES (*Performance, Information, Economy, Control, Efficiency* dan *Services*).

1. *Performance* (kinerja) adalah suatu kemampuan sistem dalam menyelesaikan tugas dengan cepat sehingga sasaran dapat segera tercapai. Kinerja diukur dengan jumlah produksi (*throughput*) dan waktu yang digunakan untuk menyesuaikan perpindahan pekerjaan *(response time*).
2. *Information* (informasi) merupakan hal penting karena dengan informasi tersebut pihak manajemen dapat melakukan pengambilan keputusan dan pengguna dapat melakukan langkah selanjutnya. Apabila kemampuan sistem baik, maka pengguna akan mendapatkan informasi yang akurat, tepat waktu dan relevan sesuai dengan yang diharapkan.
3. *Economy* (ekonomi) merupakan penilaian sistem atas penghematan dan keuntungan yang akan didapatkan dari sistem yang dikembangkan. Peningkatan terhadap kebutuhan ekonomis mempengaruhi pengendalian biaya dan peningkatan manfaat.
4. *Control* (kendali) dipasang untuk meningkatkan kinerja sistem, mencegah atau mendeteksi kesalahan sistem, menjamin keamanan data, informasi dan kebutuhan.

5. *Efficiency* (efisiensi) menyangkut bagaimana menghasilkan *output* sebanyak-banyaknya dengan *input* yang sekecil mungkin. Sistem dapat dikatakan tidak efisien bila banyak waktu yang terbuang pada aktivitas sumber daya manusia, mesin dan komputer, peng-*input*-an data yang berlebihan, pemrosesan data yang berlebihan, atau informasi yang dihasilkan berlebihan.
6. *Service* (pelayanan) menyangkut penilaian dari suatu sistem yang dilihat pula dari kriteria-kriteria seperti keakuratan dan konsistensi produk yang dihasilkan sistem, kemudahan sistem untuk dipelajari dan digunakan, atau fleksibilitas.

2.7 *Visual Basic For Application* (VBA)

VBA menurut MacDonald (2010) adalah bahasa pemrograman untuk *Microsoft Office* dan aplikasi yang terkait. Dalam VBA juga dapat menggunakan *macro* untuk membuat objek dalam aplikasi sistem yang koheren. Dengan menggunakan *macro* dapat membantu menyelesaikan tugas – tugas kecil dalam aplikasi *Microsoft.* VBA digunakan kerana beberapa alasan, diantaranya adalah:

1. Membuat aplikasi lebih mudah dipertahankan.
2. Dapat membuat fungsi sesuai keinginan.
3. Membuat atau memanipulasi objek.
4. Melakukan sistem berbasis pada tindakan.
5. Memanipulasi *record* pada suatu waktu.