

BAB II

TINJAUAN PUSTAKA

2.1 Sistem Informasi

Sistem informasi adalah satu sistem yang di dalam suatu organisasi yang mempertemukan kebutuhan pengelola transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan. [1]

Sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building blocks*) yaitu: [1]

1. Blok Masukan (*Input Block*)

Input mewakili data yang masuk kedalam sistem informasi. *Input* disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan yang dapat berupa dokumen-dokumen dasar.

2. Blok Model (*Model Block*)

Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data *input* dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran (*Output Block*)

Produk dari sistem informas adalah keluaran yang merupakan informasi yang sberkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen dan semua pemakai sistem.

4. Blok Teknologi (*Technology Block*)

Teknologi merupakan “kotak alat” dalam sistem informasi. Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran, serta membantu pengendalian dari sistem secara keseluruhan.

5. Blok Basisdata (*Database Block*)

Basisdata merupakan kumpulan dari data yang saling berhubungan serta dengan lainnya, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak

untuk memanipulasi. Data perlu disimpan didalam basisdata untuk keperluan penyediaan informasi lebih lanjut.

2. Blok Kendali (*Control Block*)

Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila sudah terlanjur, masalah dapat segera diatasi.

2.2 Administrasi

Ada dua pengertian administrasi, yaitu administrasi dalam arti sempit dan administrasi dalam arti luas. [2]

1. Administrasi dalam arti sempit adalah kegiatan penyusunan dan pencatatan data dan informasi secara sistematis dengan tujuan untuk menyediakan keterangan serta memudahkan memperolehnya kembali secara keseluruhan dan dalam satu hubungan satu sama lain. Administrasi dalam arti sempit ini sebenarnya lebih tepat disebut dengan tata usaha.
2. Administrasi dalam arti luas adalah kegiatan kerja sama yang dilakukan sekelompok orang berdasarkan pembagian kerja sebagaimana ditentukan dalam struktur dengan mendayagunakan sumber daya untuk mencapai tujuan secara efektif dan efisien. Jadi, pengertian administrasi dalam arti luas memiliki unsur-unsur sekelompok orang. Kerja sama pembagian tugas secara terstruktur. Kegiatan yang runtut dalam proses, tujuan yang akan dicapai, dan pemanfaatan berbagai sumber.

Ada lima unsur utama dalam administrasi yaitu [3]:

- a. Adanya kelompok yang terdiri dari dua orang atau lebih
- b. Adanya kerja sama dari kelompok tersebut
- c. Adanya kegiatan atau proses usaha
- d. Adanya bimbingan, kepemimpinan dan pengawasan
- e. Adanya tujuan

Administrasi kesehatan adalah aplikasi dan teknik administrasi yang dilakukan dalam praktik kesehatan masyarakat dan ilmu sosial dalam kegiatan masyarakat yang terorganisir dan dirancang untuk melindungi serta meningkatkan kesehatan individu, kelompok, masyarakat atau institusi. Tiga unsur pokok administrasi kesehatan adalah:

- a. Aplikasi fungsi administrasi
- b. Teknik, perangkat atau piranti administrasi
- c. Pencapaian tujuan administrasi kesehatan dalam pemenuhan tuntutan dan kebutuhan derajat kesehatan masyarakat setinggi-tingginya [3].

2.3 Sistem Pelayanan Kesehatan

Pelayanan kesehatan adalah upaya yang diselenggarakan sendiri atau secara bersama-sama dalam suatu organisasi untuk memelihara dan meningkatkan kesehatan, mencegah dan menyembuhkan penyakit serta memulihkan kesehatan perorang, keluarga, kelompok ataupun masyarakat [4].

Ada tiga tingkatan pelayanan kesehatan, antara lain:

1. Pelayanan kesehatan tingkat pertama adalah pelayanan kesehatan perorangan yang bersifat umum yang meliputi pelayanan rawat jalan tingkat pertama dan rawat inap tingkat pertama.
2. Pelayanan kesehatan tingkat lanjutan adalah upaya pelayanan kesehatan perorangan yang bersifat spesialisik atau sub spesialisik yang meliputi rawat jalan tingkat lanjutan dan rawat inap tingkat lanjutan diruang perawatan khusus.
3. Pelayanan kesehatan penunjang adalah pelayanan yang diberikan untuk menunjang pelayanan kesehatan, yang meliputi pelayanan obat, pemeriksaan penunjang diagnostik dan pelayanan penunjang lainnya [4].

2.3.1 Klinik

Klinik merupakan tempat pelayanan kesehatan yang hampir sama dengan rumah sakit, tetapi fasilitas medisnya lebih sederhana. Terhadap klinik pertama yang dipimpin dokter umum atau dokter gigi umum dan berwenang melakukan layanan medis dasar. Tingkat yang lebih disebut klinik utama, yang dipimpin seorang dokter spesialis atau seorang dokter gigi spesialis. Di klinik utama, pelayanan medis spesialis bisa dilakukan atau khusus untuk melayani keluhan tertentu. Biasanya klinik tersebut dijalankan oleh dokter-dokter yang menjalankan praktik pribadi secara berkelompok. Klinik biasanya hanya menerima rawat jalan. Namun, klinik utama dapat pula dilengkapi dengan fasilitas rawat inap dengan memenuhi persyaratan yang lebih lengkap (disebut klinik rawat inap). Klinik dapat dioperasikan, dikelola dan didanai

secara pribadi ataupun publik dan meliputi perawatan kesehatan dasar maupun spesialisik dan bisa dilengkapi dengan fasilitas *home care* [5].

Klinik adalah fasilitas pelayanan kesehatan yang menyelenggarakan dan menyediakan layanan medis dasar dan atau spesialisik diselenggarakan oleh lebih dari satu jenis tenaga kesehatan dan dipimpin oleh seorang tenaga medis (Permenkes RI/No.9.2014). Klinik dibagi menjadi dua yaitu (Permenkes RI/No.9.2014):

1. Klinik Pratama, merupakan klinik yang menyelenggarakan pelayanan medik dasar (bisa dibentuk oleh perorangan dan badan usaha)
2. Klinik Utama, merupakan klinik yang menyelenggarakan pelayanan medik spesialisik atau pelayanan medik dasar dan spesialisik (harus berbentuk badan hukum).

2.3.2 Rawat Inap

Manfaat rawat inap adalah penggantian biaya pengobatan dan perawatan untuk sakit yang membutuhkan rawat inap (opname). Penggantian yang dijamin oleh manfaat rawat inap antara lain biaya akomodasi ruangan termasuk unit rawat intensif, jasa kunjungan dokter ruangan termasuk dokter spesialis, biaya tindakan bedah termasuk jasa dokter, dokter anestesi juga jasa kamar operasi [5].

Pasien memerlukan perawatan, dapat dibagi menjadi tiga kelompok yaitu [4]:

1. Pasien yang tidak urgen, penundaan perawatan pasien tersebut tidak akan menambah penyakitnya.
2. Pasien urgen, tetapi tidak gawat darurat, dapat dimasukkan ke dalam daftar tunggu.
3. Pasien gawat darurat (*emergency*) langsung dirawat. Pasien urgen tetapi tidak darurat dan pasien yang tidak urgen.

Prosedur selama di ruang perawatan. Pada waktu pasien tiba diterima oleh perawat, pasien diberi tanda pengenal. Perawat menambah formulir yang diperlukan. Selama perawatan, perawat mencatat semua data perawatan yang diberikan dari mulai pasien tiba di ruang sampai pasien tersebut pulang, dipindahkan atau meninggal.

2.3.3 Rawat Jalan

Yang dimana dengan rawat jalan dan pelayanan rawat jalan, yaitu (Kepmenkes No.66/MENKES/11/1987):

- a. Rawat jalan adalah pelayanan terhadap orang yang masuk ke rumah sakit /puskesmas/klinik, untuk keperluan observasi, diagnosis, pengobatan, rehabilitasi medik dan pelayanan kesehatan lainnya tanpa tinggal diruang inap.
- b. Pelayanan rawat jalan adalah pelayanan yang diberikan di unit pelaksanaan fungsional rawat jalan terdiri dari poliklinik umum dan poliklinik spesialis serta unit gawat darurat.

Manfaat rawat jalan adalah penggantian biaya pengobatan di klinik atau rumah sakit yang tidak memerlukan penginapan. Dalam manfaat ini, biaya yang diganti asuransi antara lain honor dokter, tindakan diagnostik, biaya pemeriksaan laboratorium dan foto *rontgen*. Beberapa perusahaan asuransi memperluas manfaat rawat jalan dengan program persalinan (kelahiran) [5].

Penerimaan pasien rawat jalan yaitu [6]:

1. Pasien baru

Setiap pasien baru diterima di Tempat Penerimaan Pasien (TPP) dan akan di wawancarai oleh petugas guna mendapatkan informasi mengenai data identitas sosial pasien yang harus diisikan pada formulir ringkasan riwayat klinik. Setiap pasien baru akan memperoleh nomor pasien yang akan digunakan sebagai kartu pengenal (kartu berobat), yang harus dibawa pada setiap kunjungan berikutnya ke rumah sakit yang sama, baik sebagai pasien berobat jalan maupun sebagai pasien rawat inap.

Pada rumah sakit yang telah menggunakan sistem komputerisasi identitas sosial pasien yang disertai nomor rekam medis pasien baru harus disimpan untuk dijadikan *database* pasien, sehingga sewaktu-waktu pasien berobat kembali kerumah sakit maka data pasien tersebut akan mudah ditemukan dengan cepat. Data pada ringkasan riwayat klinik diantaranya berisi: dokter, penanggung jawab poliklinik, nomor pasien, alamat lengkap, tempat/tanggal lahir, umur, jenis kelamin, status keluarga, dan agama dan pekerjaan.

Ringkasan riwayat klinik ini juga dipakai sebagai dasar pembuatan. Kartu Indeks Utama Pasien (KIUP) dan data diatas pula yang disimpan sebagai *database* bagi rumah sakit yang telah menggunakan sistem komputerisasi. Setelah selesai dalam proses pendaftaran, pasien baru dipersilahkan menunggu dipoliklinik yang dituju dan petugas rekam medis mempersiapkan berkas rekam medisnya lalu dikirim ke poliklinik tujuan pasien. Setelah mendapatkan pelayanan yang cukup dari poliklinik. Ada beberapa kemungkinan dari setiap pasien. Pasien boleh langsung pulang. Pasien diberi slip perjanjian oleh petugas poliklinik untuk datang kembali pada hari dan tanggal yang telah ditetapkan. Kepada pasien diminta datang kembali, harus lapor kembali ke TPP. Pasien dirujuk/dikirim kerumah sakit lain atau Pasien harus keruang perawatan. Semua berkas rekam medis pasien poliklinik yang telah selesai berobat harus kembali ke Instalasi Rekam Medis, kecuali pasien pasien yang harus dirawat, rekam medisnya akan dikirim keruang perawatan.

2. Pasien Lama

Untuk pasien lama atau pasien yang pernah datang berobat sebelumnya ke rumah sakit, maka pasien mendatangi tempat pendaftaran pasien lama atau tempat penerimaan pasien yang telah ditentukan. Pasien lama ini dapat dibedakan. Pasien yang datang dengan perjanjian pasien yang datang tidak dengan perjanjian (atas kemauan sendiri). Baik pasien dengan perjanjian maupun pasien yang datang atas kemauan sendiri, setelah memberi karcis, baru akan mendapatkan pelayanan di TPP. Pasien perjanjian akan langsung menuju poliklinik yang dituju karena rekam medisnya telah disiapkan oleh petugas. Sedangkan untuk pasien yang datang atas kemauan sendiri/bukan pasien perjanjian, harus menunggu sementara rekam medisnya dimintakan oleh petugas TPP ke Instalasi Rekam Medis tersebut dikirim ke poliklinik oleh petugas, selanjutnya pasien akan mendapatkan pelayanan kesehatan di poliklinik dimaksud.

2.3.4 BPJS (Badan Penyelenggara Jaminan Kesehatan)

BPJS adalah badan hukum publik yang berfungsi untuk menyelenggarakan program jaminan sosial bagi seluruh Republik Indonesia.

Peserta BPJS kesehatan ada 2 kelompok yaitu [4]:

1. PBI Jaminan Kesehatan

Pemerintah Bantuan Iuran (PBI) adalah peserta jaminan kesehatan bagi fakir miskin dan orang yang tidak mampu sebagaimana dimanakan UU SJSN yang iurannya dibayar pemerintah sebagai peserta program jaminan kesehatan. Peserta PBI adalah fakir miskin yang ditetapkan oleh pemerintah dan diatur melalui peraturan pemerintah. Selain fakir miskin, yang berhak menjadi peserta PBI Jaminan kesehatan lainnya adalah yang mengalami cacat total tetap dan tidak mampu.

2. Bukan PBI Jaminan Kesehatan

Peserta bukan PBI Jaminan Kesehatan terdiri atas:

a. Pekerja penerima upah dan anggota keluarganya

Pekerja penerima upah adalah setiap orang yang bekerja pada pemberi kerja dengan menerima gaji atau upah, seperti Pegawai Negeri Sipil (PNS), anggota TNI, anggota POLRI, pejabat negara, pegawai pemerintah non pegawai negeri (pegawai honorer, staff khusus, staff ahli), pegawai swasta, dan pekerja lain yang memenuhi kriteria pekerja penerima upah.

b. Pekerja bukan penerima upah dan anggota keluarganya

Pekerja bukan penerima upah adalah setiap orang yang bekerja atau berusaha atas risiko sendiri, seperti pekerja diluar hubungan kerja atau pekerja mandiri atau pekerja lain yang memenuhi kriteria pekerja bukan penerima upah.

- c. Bukan pekerja dan anggota lainnya

Sedangkan yang termasuk kategori bukan pekerja adalah investor, pemberi kerja, penerima pensiun, dan bukan pekerja lain yang memenuhi kriteria bukan pekerja penerima upah.

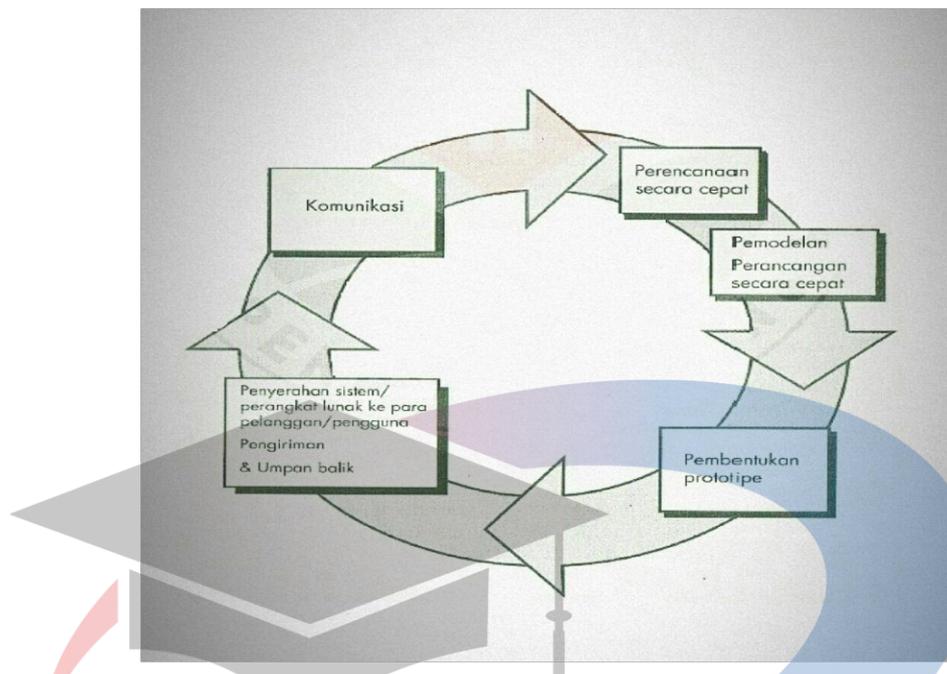
2.4 Metode *Prototyping*

Prototyping merupakan teknik pengembangan sistem yang menggunakan *prototype* untuk menggambarkan sistem, sehingga pengguna atau pemilik sistem mempunyai gambaran pengembang sistem yang akan dilakukannya. Teknik ini sering digunakan apabila pemilik sistem tidak terlalu menguasai sistem yang akan dikembangkannya, sehingga pemilik memerlukan gambaran dari sistem yang akan dikembangkannya tersebut.

Dengan teknik *prototyping*, pengembang bisa membuat *prototype* terlebih dahulu sebelum mengembangkan sistem yang sebenarnya. Contoh dari *prototype* misalnya, dalam dunia *automotif*, ketika *vendor* dari suatu mobil ingin mengembangkan sistem dari mobil yang sudah diproduksinya, mereka sering membuat *prototype* sebagai gambaran untuk mewakili sistem yang sebenarnya, dan tak jarang ketika sistem tersebut sudah akan diluncurkan dipasaran, para calon konsumen pun diberikan *prototype* tersebut, agar calon konsumen mempunyai gambaran dari mobil yang akan dibelinya. *Prototype* dari sistem mobil ini bisa berupa mobil kecil yang berbentuk seperti mainan. Dalam pengembangan sistem informasi, *prototype* sering diwujudkan dalam bentuk *user interface* program aplikasi dan contoh *reporting* yang akan dihasilkan, sehingga dengan demikian pengguna sistem akan mempunyai gambaran tentang sistem yang akan digunakan nanti [7].

2.4.1 Tahapan *Prototyping*

Adapun tahapan-tahapan metode *prototype* dapat didefinisikan sebagai berikut [8]:



Gambar 2. 1 Metode *Prototyping*

Pembuatan *prototype* dimulai dengan dilakukannya komunikasi antara tim pengembang perangkat lunak dengan para pelanggan. Tim pengembang perangkat lunak akan melakukan pertemuan-pertemuan dengan para *stakeholder* untuk mendefinisikan sasaran keseluruhan untuk perangkat lunak yang akan dikembangkan, mengidentifikasi spesifikasi kebutuhan apapun yang saat ini diketahui, dan menggambarkan area-area dimana definisi lebih jauh pada iterasi selanjutnya merupakan keharusan. Iterasi pembuatan *prototype* direncanakan dengan cepat dan pemodelan (dalam bentuk “rancangan cepat”) dilakukan. Suatu rancangan cepat berfokus pada representasi semua aspek perangkat lunak yang akan terlihat oleh pengguna akhir (misalnya rancangan antarmuka pengguna [*user interface*] atau format tampilan). Rancangan cepat (*quick design*) akan memulai konstruksi pembuatan *prototype*. *Prototype* kemudian akan diserahkan kepada para *stakeholder* dan kemudian mereka akan melakukan evaluasi-evaluasi tertentu terhadap *prototype* yang telah dibuat sebelumnya, kemudian akhirnya akan memberikan umpan balik yang akan digunakan untuk memperhalus spesifikasi kebutuhan. Iterasi akan terjadi saat *prototype* diperbaiki untuk memenuhi kebutuhan dari para *stakeholder*, sementara pada saat yang sama memungkinkan kita lebih memahami kebutuhan apa yang akan dikerjakan pada iterasi selanjutnya [8].

Tahapan dalam model *prototyping* yaitu [8]:

1. Komunikasi terlebih dahulu yang dilakukan antara pelanggan dengan tim pengembang perangkat lunak mengenai spesifikasi kebutuhan yang diinginkan.
2. Akan dilakukan perencanaan dan pemodelan secara cepat berupa rancangan cepat (*quick design*) dan kemudian akan memulai konstruksi pembuatan *prototype*.
3. *Prototype* kemudian akan diserahkan kepada para *stakeholder* untuk dilakukan evaluasi lebih lanjut sebelum diserahkan kepada para pembuat *software*.
4. Pembuatan *software* sesuai dengan *prototype* yang telah dievaluasi yang kemudian akan diserahkan kepada pelanggan.
5. Jika belum memenuhi kebutuhan dari pelanggan maka akan kembali ke proses awal sampai dengan kebutuhan dari pelanggan telah terpenuhi.

2.4.2 Kelebihan dan Kelemahan Model *Prototyping*

Adapun kelemahan dan kelebihan dari model *prototyping* sebagai berikut [9]:

Kelemahan *prototyping*

1. Kelemahan *prototyping*

Sama halnya dengan teknik pengumpulan data lainnya, ada beberapa kelemahan dalam melakukan *prototyping*. Yang pertama ialah, sulitnya mengatur *prototyping* sebagai suatu proyek dalam sistem yang lebih besar. Kedua, pengguna dan penganalisis bisa-bisa mengadopsi prototipe sebagai suatu sistem yang komplet bila pada kenyataannya sistem tersebut tidak cukup memadai dan tidak pernah dimaksudkan sebagai suatu sistem yang sudah jadi.

2. Kelebihan *prototyping*

Prototipe tidak diperlukan atau tidak sesuai untuk setiap proyek sistem, seperti yang kita lihat. Namun kelebihan-kelebihan yang dimilikinya juga perlu dipertimbangkan saat memutuskan segala sesuatunya terhadap prototipe. Ada tiga kelebihan utama *prototyping* yang potensial untuk mengubah sistem lebih dini dalam masa perkembangannya, peluang untuk menghentikan pengembangan suatu sistem yang tidak berfungsi dan kemungkinan mengembangkan suatu sistem yang lebih mendekati apa yang dibutuhkan dan diharapkan pengguna. Ketiga kelebihan tersebut saling berkaitan satu sama lain.

2.4.3 Jenis-jenis *Prototipe*

Kata prototipe banyak digunakan untuk berbagai macam hal. Dari pada berupaya menyatukan semua kegunaan ini kedalam satu definisi atau berusaha memerintahkan satu pendekatan yang benar terhadap suatu topik *prototyping* yang kontroversial, kami menggambarkan bagaimana masing-masing dari beberapa konsepsi *prototyping* sangat berguna bila diterapkan dalam suatu situasi tertentu [10].

1. Prototipe *Patched-up*

Jenis *prototyping* yang pertama ini berkaitan dengan penyusunan sistem yang bekerja namun *patch* atau *patch*-bersama-sama. Contohnya dalam sistem informasi adalah model kerja yang memiliki semua fitur yang diperlukan namun tidak efisien. Dalam hal *prototyping* ini, pengguna dapat berinteraksi dengan sistem sesuai dengan antarmukanya serta jenis keluaran yang tersedia. Pencarian dan penyimpanan informasi bisa menjadi sangat tidak efisien, karena program-programnya ditulis secara cepat agar bisa berfungsi, atau lebih dari sekedar efisien. Contoh lain untuk prototipe *patch-up* adalah suatu sistem informasi yang memiliki semua prototipe yang diajukan, tetapi menjadi model dasar yang sebenarnya akan ditingkatkan.

2. Prototipe Non-operasional

Konsepsi prototype kedua adalah prototipe dari model skala nganggur yang disusun untuk menguji beberapa rancangan tertentu. Satu model skala nganggur dari suatu sistem informasi bisa diproduksi bila pengkodean yang diperlukan oleh aplikasi terlalu mahal untuk prototipe kecuali bila ada gagasan yang dianggap sangat bermanfaat dari sistem bisa diperoleh melalui *prototyping* masukan dan keluaran saja. Mengenai pemrosesan ini, karena masalah biaya dan waktu, tidak akan diprototipekan. Beberapa keputusan mengenai kegunaan sistem masih dapat dibuat berdasarkan masukan dan keluaran yang diprototipekan.

3. Prototipe *First-of-Series*

Konsepsi *prototyping* ketiga melibatkan penciptaan suatu model skala lengkap pertama dari sistem, yang disebut pilot. Jenis *prototyping* semacam ini digunakan ketika merencanakan banyak versi dari sistem informasi yang sama. Model kerja skala lengkap ini memungkinkan pengguna bisa mendapat pengalaman interaksi yang realistis dengan sistem baru, yang sebelumnya meminimalkan biaya untuk

menanggulangi masalah-masalah yang ada. Contohnya ditemukan dalam instalasi perbankan untuk pentransferan dana secara elektronik. Pertama-tama, dipasang sebuah prototipe berskala lengkap disalah satu dari dua lokasi yang ada, dan apabila berhasil, dipasang duplikat-duplikatnya disemua lokasi berdasarkan pola-pola yang digunakan konsumen serta faktor-faktor penting lainnya.

4. Prototipe Fitur-fitur Terpilih

Konsepsi *prototyping* keempat berkaitan dengan pembangunan suatu model operasional yang mencakup beberapa, tapi tidak semua, fitur-fitur yang dimiliki sistem final.

Analoginya, ibarat sebuah pusat perbelanjaan retail baru yang akan dibuka sebelum pembangunan seluruh bagian toko selesai. Pada pusat retail yang baru dibangun tersebut, fungsi utamanya ialah agar orang bisa membeli beberapa barang tertentu. Bila melakukan *prototyping* sistem informasi dengan cara ini, meski tidak semua, fitur-fitur intinya harus dimasukkan. Bila *prototyping* jenis ini dilakukan, sistem akan bisa tercapai sesuai modul sehingga bila fitur-fitur yang dibuat prototipenya dianggap berhasil, maka akan dipadukan ke dalam sistem final yang besar tanpa melakukan pekerjaan yang besar dalam antarmuka.

2.4.4 Petunjuk-petunjuk Untuk Mengembangkan Sebuah *Prototipe*

Sekali keputusan untuk prototipe dibuat, ada 4 petunjuk yang harus diamati saat mengintegrasikan *prototyping* kedalam fase penetapan syarat-syarat SHPS, yakni [10]:

1. Bekerja sesuai modul

Saat membuat prototipe beberapa fitur dari sebuah sistem menjadi model yang bisa berfungsi dengan baik, tidak boleh tidak penganalisis bekerja menurut modul yang bisa dilaksanakan. Salah satu kelebihan *prototyping* yang benar-benar berbeda ialah dimana *prototyping* tidak diperlukan atau diharapkan mampu membangun sistem kerja secara keseluruhan untuk tujuan-tujuan tertentu yang berkaitan dengan sistem.

2. Membangun Prototipe dengan cepat

Penganalisis dapat menggunakan *prototyping* untuk mempersingkat kesenjangan ini dengan menggunakan teknik pengumpulan informasi tradisional untuk

menentukan syarat-syarat informasi yang penting, sehingga kemudian mereka dapat membuat keputusan dengan cepat yang membantu model kerja.

3. Memodifikasi Prototipe

Petunjuk ketiga mengembangkan prototipe ialah bahwa penyusunannya harus mendukung dilakukannya modifikasi-modifikasi tertentu. Membuat prototipe yang bisa dimodifikasi berarti menciptakannya dalam modul-modul yang sangat tidak saling tergantung. Bila petunjuk ini diamati, maka akan ditemui berkurangnya daya tahan saat modifikasi dalam prototipe diperlukan.

4. Menekankan antarmuka pengguna

Antarmuka pengguna dengan prototipe (dan sistem yang sebenarnya) sangat penting. Karena apa yang sedang benar-benar Anda upayakan untuk dicapai dengan prototipe adalah agar pengguna menyatakan syarat-syarat informasi mereka, mereka harus mampu berinteraksi dengan mudah dengan prototipe sistem. Untuk sebagian besar pengguna, antarmuka adalah sistem. Jadi tidak harus menjadi blok-blok penghalang.

2.4.5 Daya Tarik *Prototyping*

Pengguna maupun pengembang menyukai *prototyping* karena alasan-alasan dibawah ini [9]:

1. Membaiknya komunikasi antar pengembang dan pengguna.
2. Pengembang dapat melakukan pekerjaan yang lebih baik dalam menentukan kebutuhan pengguna.
3. Pengguna memainkan peranan yang lebih aktif dalam pengembangan sistem.
4. Pengembang dan pengguna menghabiskan waktu dan usaha yang lebih sedikit dalam mengembangkan sistem.
5. Implementasi menjadi jauh lebih mudah karena pengguna tau apa yang diharapkannya.

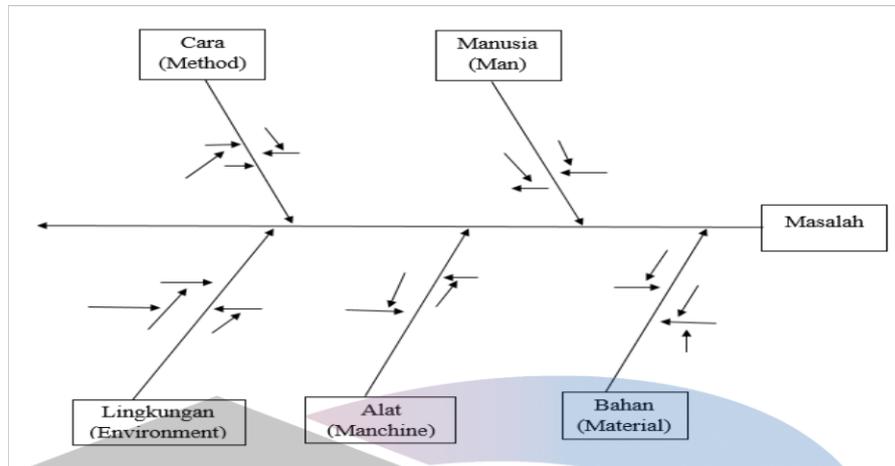
2.5 Diagram Ishikawa/*Fishbone*

Diagram tulang ikan (*fishbone* diagram) atau ishikawa diagram (sesuai dengan nama penemunya, Dr. Kaoru Ishikawa), adalah alat analisis yang memberi cara pandang yang sistematis terhadap sebab dan akibat yang timbul atau memiliki

kontribusi pada suatu akibat. Karena fungsi inilah *fishbone* diagram juga disebut sebagai *cause-effect* diagram. Apa pun nama yang dipilih, yang harus diingat adalah bahwa diagram ini berguna untuk membantu melakukan kategorisasi dari penyebab potensial dari suatu masalah atau isu dan mengidentifikasi penyebab utama.

Langkah-langkah untuk membuat *fishbone* diagram adalah sebagai berikut [11]:

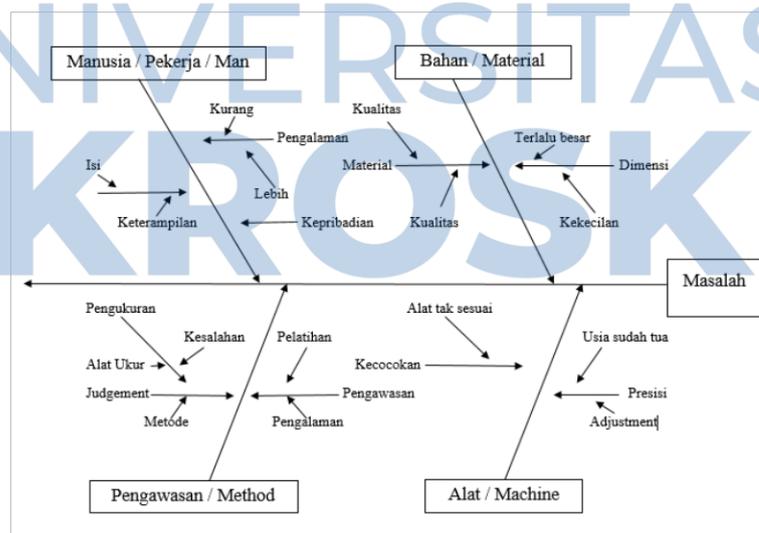
1. Gambarkan diagram tulang ikan.
2. Buat daftar masalah/isu yang dipelajari pada “kepala ikan”.
3. Berikan label pada tiap-tiap “tulang”. Kategori utama yang biasa dipakai adalah:
 - a. 4M (*Method, Machine, Material, Manpower*)
 - b. 4P (*Place, Procedure, People, Policies*)
 - c. 4S (*Surrounding, Supplier, System, Skill*). Kategori tersebut dapat dikombinasikan untuk memperkaya ide dan membantu dalam pengorganisasian ide.
4. Gunakan teknik *idea-generating*, misalnya: *brainstorming* untuk mengidentifikasi faktor pada tiap kategori yang mungkin menyebabkan masalah/isu dan atau akibat yang sedang dihadapi. Tim seharusnya memperkaya ide dengan memunculkan pertanyaan seperti: “apa yang menyebabkan /mengakibatkan masalah yang sedang dihadapi...”
5. Ulangi prosedur diatas untuk masing-masing faktor kategori yang menghasilkan sub-faktor. Lanjutkan dengan pertanyaan seperti: “mengapa ini terjadi.” Dan tambahkan segmen tambahan pada tiap faktor juga tiap sub-faktor.
6. Lanjutkan sampai tidak ada lagi informasi penting saat timbul pertanyaan “Mengapa ini terjadi.”
7. Analisa hasil dari *fishbone* diagram setelah anggota tim menyetujui bahwa jumlah yang tepat telah ditambahkan dan menggambarkan secara detail pada tiap kategori/sub-kategori. Jika terdapat beberapa hal yang sepertinya berulang pada kategori yang lain ini disebut sebagai “penyebab yang paling utama.”
8. Untuk masing-masing item yang masuk dalam kategori “penyebab yang paling umum”, tim harus meraih konsensus dalam daftar menurut prioritas, dimana priotitas pertama adalah “penyebab paling mungkin”. Berikut adalah contoh gambar diagram *Fishbone*.



Gambar 2. 2 Diagram *Fishbone*

Diagram tulang ikan dapat dipakai secara tersendiri dalam mencari pemecahan masalah, akan tetapi biasanya diagram ini digunakan bersama-sama dengan alat-alat statistik lainnya. Bagaimanapun juga, sebaiknya pada waktu menentukan pilihan faktor-faktor penyebab apa yang kemungkinan besar merupakan faktor yang paling berpengaruh terhadap timbulnya masalah. Sedapat mungkin dan sejauh mungkin dilakukan pengujian melalui alat-alat statistik lainnya.

Contoh-contoh lain untuk menjelaskan lebih jauh diagram tulang ikan:



Gambar 2. 3 Diagram *Fishbone*

2.6 Basis Data

Sebuah basis data adalah sebuah kumpulan data yang saling berhubungan secara logis, dan merupakan sebuah penjelasan dari data tersebut, yang didesain untuk menemukan data yang dibutuhkan oleh sebuah organisasi. Di dalam basis data, semua data diintegrasikan dengan menghindari duplikasi data. Basis data dapat digunakan oleh banyak department dan pemakai. Basis data tidak hanya memegang data operasional organisasi, tetapi juga penjelasan mengenai data tersebut. Karena alasan tersebut basis data juga dideskripsikan sebagai kumpulan data yang saling terintegrasi. Basis data juga merupakan sekumpulan elemen data terintegrasi yang secara logika saling berhubungan. Basis data mengonsolidasikan berbagai catatan yang terlebih dahulu disimpan dalam file-file terpisah kedalam satu gabungan umum elemen data yang menyediakan data untuk banyak aplikasi. Elemen data mendeskripsikan entitas-entitas dan hubungan antara entitas-entitas tersebut.

Desain basis data adalah proses membuat desain yang akan mendukung operasional dan tujuan perusahaan. Tujuan desain basis data adalah:

1. Menggambarkan relasi data antara data-data yang dibutuhkan oleh aplikasi dan *user view*.
 2. Menyediakan model data yang mendukung seluruh transaksi yang diperlukan.
 3. Menspesifikasikan desain dengan struktur yang sesuai dengan kebutuhan sistem.
- Ada beberapa pendekatan yang dapat digunakan dalam mendesain basis data, yaitu

[12]:

a. *Top-down*

Diawali dengan membuat data model. Pendekatan *Top-down* dapat diilustrasikan menggunakan *entity-relationship* (ER) model yang high level, lalu mengidentifikasi *entity*, dan *relationship* antar-*entity* organisasi. Pendekatan ini sesuai bagi basis data yang kompleks.

b. *Bottom-up*

Dimulai dari level dasar *attribute* (*property entity dan relationship*), menganalisa hubungan antar-*attribute*, mengelompokkannya dalam suatu relasi yang menggambarkan tipe *entity* dan relasi antara *entity*. Pendekatan ini sesuai bagi basis data dengan jumlah *attribute* yang sedikit.

c. *Inside-out*

Mirip seperti pendekatan *Bottom-up*, perbedaannya adalah tahap awal mengidentifikasi *major entity* lalu menguraikannya menjadi *entity* relasi-relasi dan *attribute-attribute* yang berhubungan dengan *major entity*.

d. *Mixed*

Menggunakan pendekatan *Bottom-up* dan *Top-down*.

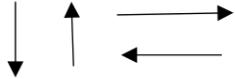
2.7 Flow of Document (FOD)

Flow of Document (FOD) adalah alat pembuatan model yang memungkinkan professional sistem untuk menggambarkan sistem sebagai satu jaringan proses fungsional yang dihubungkan satu dengan yang lainnya dengan alur data, baik secara manual maupun secara komputerisasi [6].

Berikut ini adalah symbol-simbol dari *Flow of Document (FOD)*

Tabel 2. 1 *Simbol FOD* [13]

SIMBOL	KETERANGAN
	Titik terminal, digunakan untuk menunjukkan proses awal dan akhir dari suatu proses.
	Operasi secara manual
	Fungsi pengolahan proses / menunjukkan kegiatan proses program computer
	Simbol <i>Input/output</i> , digunakan untuk mewakili data <i>input/output</i>
	Dokumen atau laporan <i>input output</i> baik untuk proses manual, mekanik atau
	Menunjukkan <i>input</i> yang menggunakan <i>on-line keyboard</i>
	Tempat penyimpanan data menggunakan <i>Harddisk</i>

	Arus informasi/kegiatan proses dari operasi program computer
	<i>File storage offline Numeric, Character</i>
	Penghubung kehalaman yang sama

2.8 Teknik Pengembangan Sistem

2.8.1 Use Case

Use case modeling / pemodelan *use case* adalah sebuah pendekatan yang memfasilitasi pengembangan berpusatkan kegunaan. *Use case modeling* adalah proses pemodelan fungsi-fungsi sistem dalam konteks peristiwa-peristiwa bisnis, siapa yang mengawalinya dan bagaimana sistem itu merespon suatu hal tersebut.

Manfaat pemodelan *use case* adalah [13]:

1. Menyediakan tool untuk meng-*capture* persyaratan fungsional.
2. Membantu menyusun ulang lingkup sistem menjadi bagian-bagian yang lebih cepat dikelola.
3. Menyediakan alat komunikasi dengan para pengguna dan *stakeholder* yang berhubungan dengan fungsionalitas sistem. *Use case* menyajikan bahasa umum yang dapat dipahami oleh berbagai macam *stakeholder*.
4. Memberikan cara bagaimana mengidentifikasi, menetapkan, melacak, mengontrol, dan mengelola kegiatan pengembangan sistem incremental dan interaktif.
5. Menyajikan panduan untuk mengsystemasi lingkup, usaha dan jadwal proyek.
6. Menyajikan garis pokok pengujian, khususnya menentukan rencana tes dan *use case*.
7. Menyajikan tool untuk melacak persyaratan.
8. Menyajikan titik mulai/awal untuk identifikasi objek data atau entitas.
9. Menyajikan spesifikasi fungsional untuk mendesain antarmuka pengguna dan sistem.
10. Menyajikan alat untuk menentukan persyaratan akses *database* dalam menambah, mengubah, menghapus dan membaca.

Use case diagram adalah *diagram* yang menggambarkan interaksi antara sistem internal dengan sistem eksternal pengguna. Dengan kata lain, secara garis besar menggambarkan siapa yang akan menggunakan sistem dan dengan cara apa pengguna mengharapkan untuk berinteraksi dengan sistem.

Use case adalah urutan langkah-langkah yang secara tindakan saling terkait baik terotomatisasi maupun secara manual untuk tujuan melengkapi satu tugas bisnis tunggal. *Actor* adalah segala sesuatu yang perlu berinteraksi dengan sistem untuk pertukaran informasi. *Association* adalah hubungan antar pelaku / *actor* dengan *use case* dimana terjadi interaksi diantara mereka. *Extension use case* adalah *use case* yang lebih kompleks untuk menyederhanakan masalah orisinal dan karena itu memperluas fungsinya. *Abstract use case* adalah *use case* yang mengurangi redundansi antara dua atau lebih *use case* dengan menggabungkan langkah-langkah yang bisa ditemukan pada *use case* tersebut [13].

2.9 PIECES (*Performance, Information, Economic, Control, Efficiency, Service*)

Proses dan teknik yang digunakan oleh analis sistem untuk mengidentifikasi, menganalisis, dan memahami persyaratan sistem disebut *requirement discovery* / penemuan persyaratan. Persyaratan sistem menentukan apa yang seharusnya dikerjakan oleh sistem informasi atau properti serta kualitas apa yang harus dimiliki oleh sistem. Persyaratan sistem yang menetapkan apa yang harus dilakukan oleh sistem informasi sering disebut persyaratan fungsional. Persyaratan sistem menetapkan properti atau kualitas yang harus dimiliki oleh sistem sering disebut persyaratan non-fungsional.

Kerangka PIECES memberikan alat unggul untuk menggolongkan persyaratan sistem. Keuntungan menggolongkan berbagai tipe persyaratan adalah kemampuan untuk menggolongkan persyaratan tersebut untuk tujuan pelaporan, pelacakan, dan validasi. Hal tersebut membantu mengidentifikasi persyaratan sistem secara cermat.

Kategori dalam kerangka PIECES yaitu [13]:

- P : Kebutuhan untuk mengoreksi atau memperbaiki *performance* / performa.
- I : Kebutuhan untuk mengoreksi atau memperbaiki *information* / informasi.
- E : Kebutuhan untuk mengoreksi atau memperbaiki *economics* / ekonomi, mengendalikan biaya atau meningkatkan keuntungan.

- C : Kebutuhan untuk mengoreksi atau memperbaiki *control* / kendali atau keamanan.
- E : Kebutuhan untuk mengoreksi atau memperbaiki *efficiency* /efesiensi orang dan proses.
- S : Kebutuhan untuk mengoreksi atau memperbaiki *service* / layanan ke pelanggan, pemasok, rekan kerja, karyawan, dan lain-lain.



UNIVERSITAS MIKROSKIL