

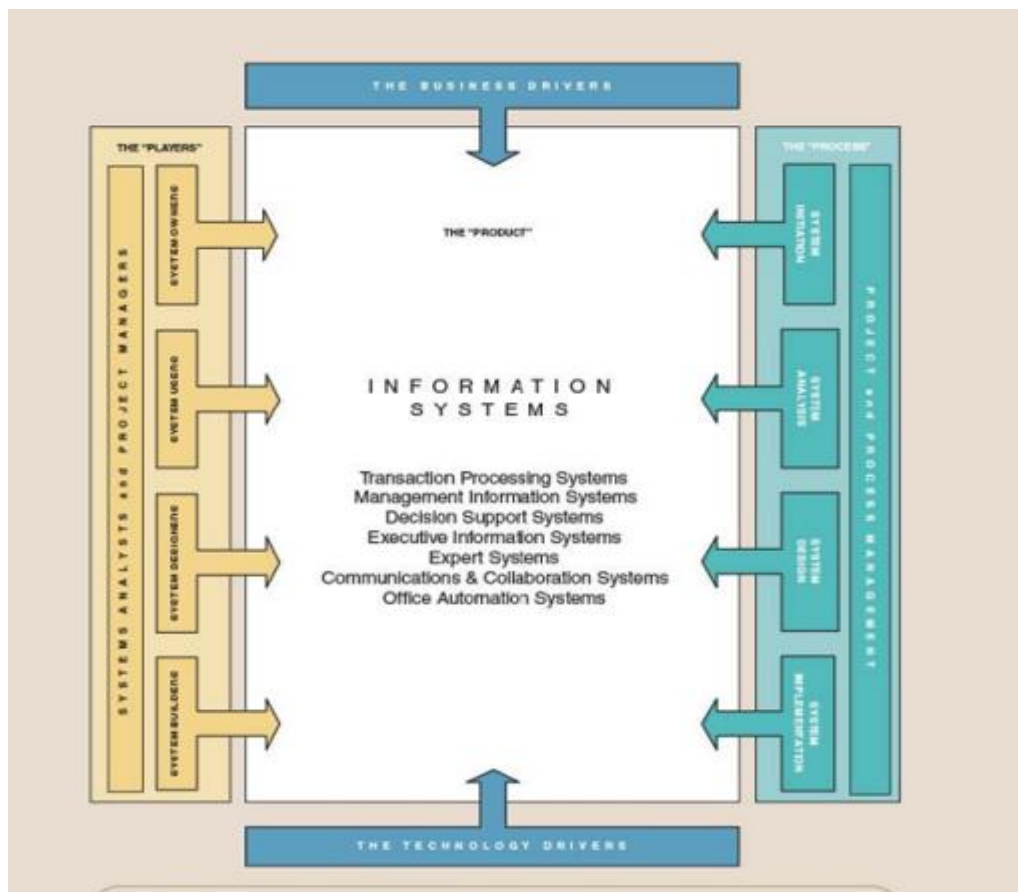
Bab 2

Tinjauan Pustaka

2.1 Sistem

Sistem adalah kumpulan dari beberapa komponen yang memiliki tujuan tertentu. Contohnya sistem akuntansi didefinisikan sebagai kumpulan dari prosedur penerimaan, pengeluaran, penjualan, pembelian dan buku besar. Sedangkan definisi lain dari sistem yaitu kumpulan dari komponen yang saling berhubungan satu dengan yang lainnya membentuk satu kesatuan untuk mencapai tujuan tertentu[2].

Berikut dibawah ini gambaran dari suatu sistem :



Gambar 2.1 Gambaran dari suatu sistem

2.2 Informasi

Informasi adalah jenis utama dari sumber daya yang didapatkan oleh seseorang dan memiliki arti sendiri bagi penggunaannya. Pengelolaan menyediakan informasi dalam bentuk tertulis maupun tidak yakni dalam bentuk tulisan. Informasi dapat berasal dari internal maupun eksternal dan digunakan untuk memecahkan suatu masalah[3].

2.3 Penggajian

Penggajian adalah suatu imbalan dari pengusaha kepada pekerja untuk suatu pekerjaan atas jasa yang telah dilakukan dan dinyatakan dalam bentuk uang ditetapkan atas dasar suatu persetujuan atau peraturan undang – undang serta dibayar atas dasar suatu perjanjian kerja antara pengusaha dan pekerja termasuk tunjangan untuk pekerja itu sendiri maupun untuk keluarga pekerja.

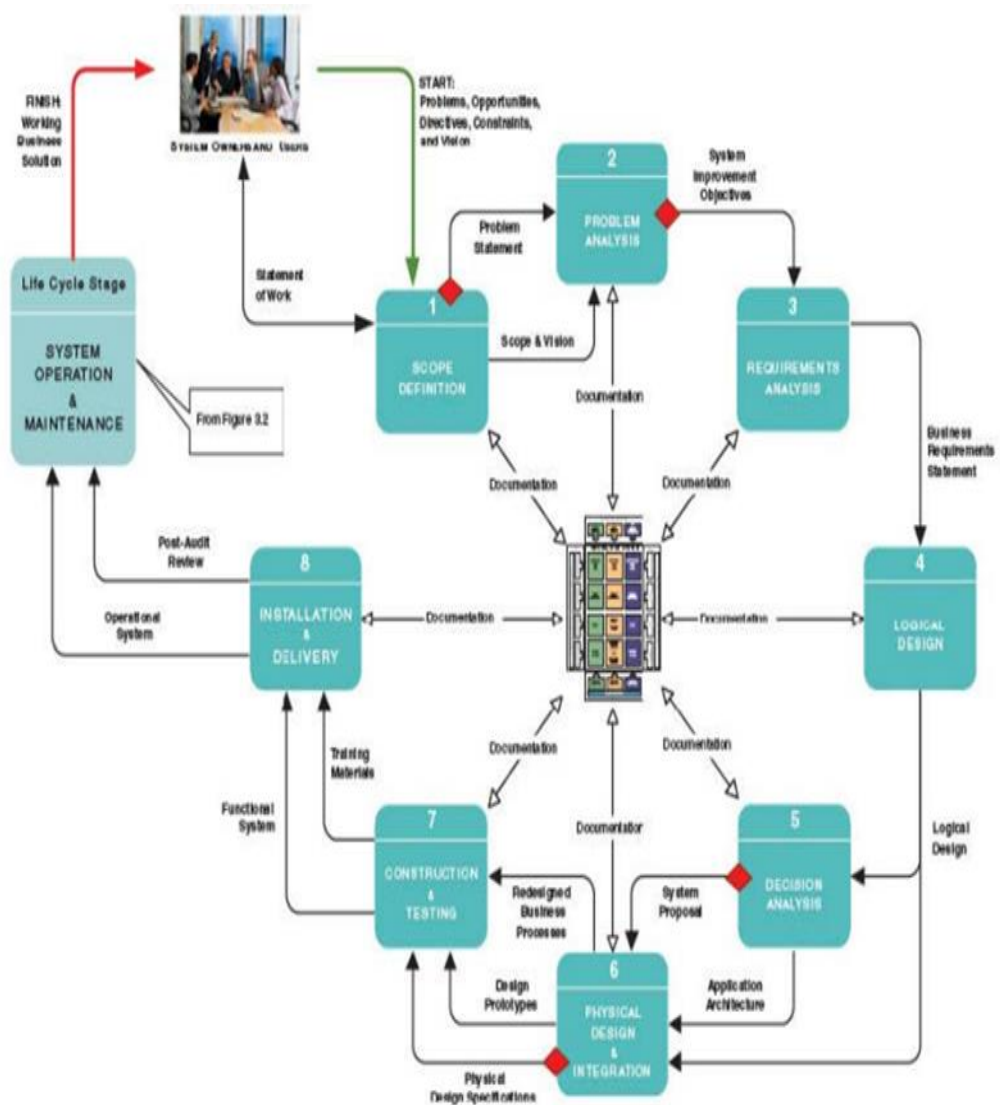
Penghasilan atau imbalan diterima oleh pekerja sehubungan dengan pekerjaan yang dilakukan pekerja dapat digolongkan dalam bentuk upah atau gaji dalam bentuk uang dan tunjangan.

Gaji pokok adalah gaji dasar yang ditetapkan untuk melaksanakan suatu jabatan atau pekerjaan tertentu pada golongan pangkat dan waktu tertentu. Gaji pokok disuatu perusahaan disusun berdasarkan jenjang jabatan dan jenjang kepangkatan [4]. Adapun pengertian Upah dalam undang-undang tenaga kerja no 13 tahun 2003, Bab 1, Pasal 1 berisikan Upah adalah hak pekerja/buruh yang diterima dan dinyatakan dalam bentuk uang sebagai imbalan dari pengusaha atau pemberi kerja kepada pekerja/buruh. MySQL banyak digunakan karena kemampuannya dalam menampung data yang cukup besar dan proses loading yang cepat dalam pemanggilan data. Database MySQL merupakan sebuah fasilitas untuk menyimpan database dan dapat mempermudah user dalam membuat form password, log in, guest book, contact, dan lain-lain.

2.4 Sistem Informasi Penggajian

Sistem informasi penggajian merupakan sistem yang menentukan, mengatur, mengawasi, mengelola data pekerja dan penggajian agar dapat memberikan data atau informasi yang cepat, tepat dan akurat oleh pihak yang berwajib [5].

Adapun diagram blok pembangunan sistem dapat dilihat pada Gambar 2.2 sebagai berikut:



Gambar 2.2 Diagram blok pembangunan sistem

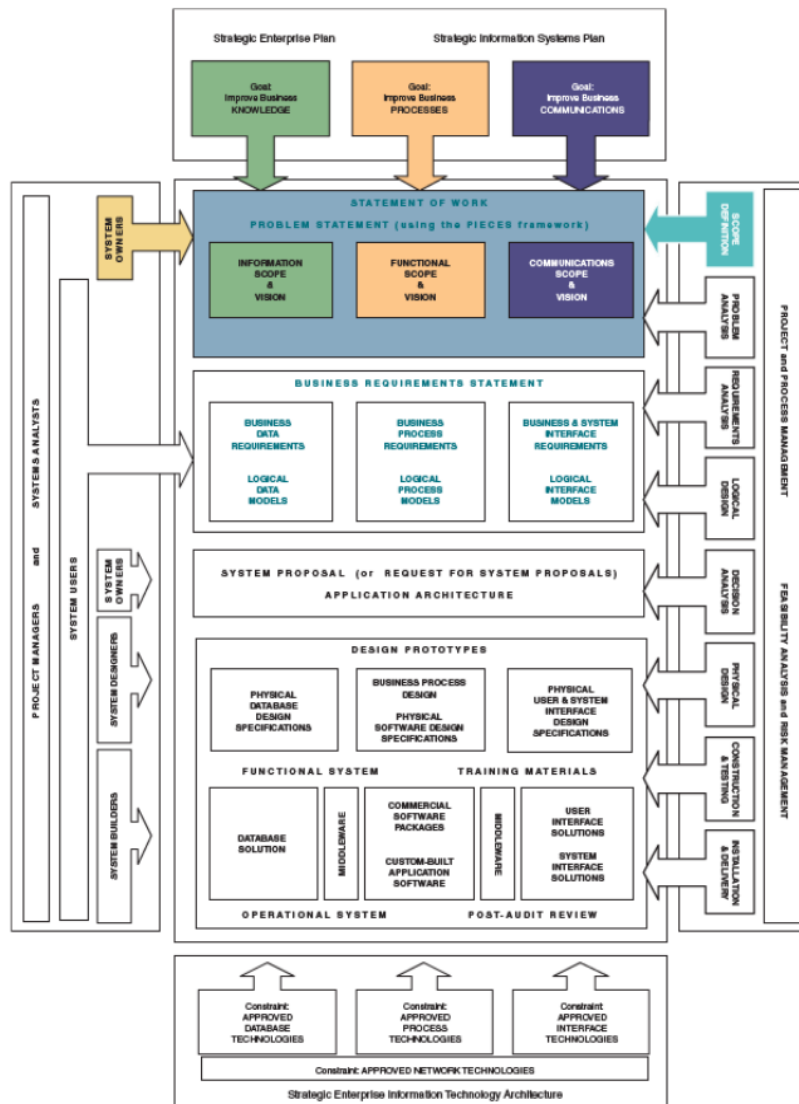
2.5 Metode FAST

Metode *FAST* adalah proses atau metodologi yang digunakan untuk memelihara dan mengembangkan sistem informasi. *FAST* memberikan kualitas sistem informasi terbaik dalam jumlah waktu wajar. Metode ini disebut juga *agile modeling* (pemodelan cerdas sejak kemampuannya untuk mendukung bukan hanya pengembangan aplikasi cepat, mendukung teknik lainnya termasuk analisis sistem yang terstruktur, teknik informasi, dan analisis berorientasi objek dan desain [6].

Metodologi *FAST* merupakan metode untuk mengembangkan sistem informasi dengan tahapan yaitu lingkup definisi (*Scope definition*), analisis permasalahan (*Problem analysis*), analisis kebutuhan (*Requirement analysis*), analisis keputusan (*Decision analysis*), desain logis (*Logical design*), desain fisik dan integrasi (*Physical design and integration*), (*construction & testing*) dan (*installation & delivery*)[6]. Menurut Whitten terdapat fase analisis sistem dalam metode *FAST*, fase-fase tersebut adalah sebagai berikut:

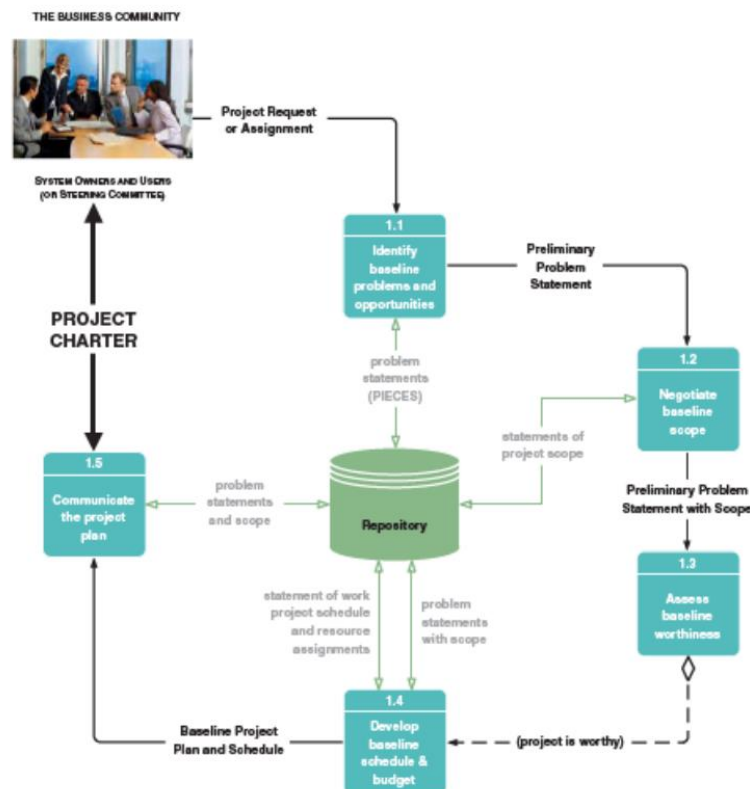
1. *Scope definition phase*

Scope definition adalah fase pertama dari proses pengembangan sistem klasik. Dalam metodologi lain ini bisa disebut *praliminary investigation phase*, *initial study phase*, *survey phase* atau fase perencanaan. Pada *Scope definition* merupakan fase yang menjawab pertanyaan, "apakah proyek ini layak untuk dilihat?". Untuk menjawab pertanyaan ini, maka harus mendefinisikan ruang lingkup proyek dan yang dirasakan bermasalah, peluang, dan arahan yang memicu proyek. Dengan asumsi proyek dianggap layak, tahap *scope definition* juga harus menetapkan rencana proyek dalam hal skala, strategi pengembangan, jadwal, kebutuhan sumber daya dan anggaran. Konteks untuk *scope definition phase* dapat dilihat pada gambar 2.3.



Gambar 2.3. Konteks *scope definition phase* pada analisis sistem

Pada fase *scope definition* terutama berkaitan dengan pngan *System Owner* tentang sistem yang ada dan masalah atau peluang yang memicu minat. *Sistem Owner* cenderung peduli dengan gambaran besar, bukan detail. Selanjutnya, yang ditentukan apakah sumber daya dapat dan akan berkomitmen untuk proyek atau tidak.



Gambar 2.4. Task untuk *scope definition phase* pada analisis sistem

Pada gambar 2.4 bagian pertama dari lima *task diagram* akan memperlihatkan cara untuk mengambil atau melihat lebih dekat pada setiap tahap analisis sistem. *Task diagram*, menunjukkan pekerjaan (tugas) yang harus dilakukan untuk menyelesaikan fase. *Task diagram* akan menjelaskan dalam paragraf yang menyertainya pendekatan, alat, dan teknik yang mungkin ingin dipertimbangkan untuk setiap tugas. Pada gambar 2.4. juga menunjukkan tugas yang diperlukan untuk *scope definition phase*. Hal ini penting untuk ingatlah bahwa diagram tugas ini hanyalah templat. Tim proyek dan proyek manajer dapat memperluas atau mengubah template ini untuk mencerminkan kebutuhan unik dari setiap diberikan proyek.

Penyampaian akhir untuk penyelidikan awal fase adalah penyelesaian piagam proyek. (Hasil utama tersebut ditunjukkan dalam setiap *task diagram* dalam huruf kapital semua.) Piagam proyek mendefinisikan ruang lingkup proyek, rencana, metodologi, dan sebagainya. Penyelesaian piagam proyek merupakan tonggak pertama dalam sebuah proyek. *Scope definition phase* dimaksudkan untuk menjadi cepat. Seluruh fase harus tidak melebihi dua atau tiga hari untuk sebagian besar proyek. Fase biasanya mencakup tugas berikut:

1. Mengidentifikasi masalah dan peluang dasar
2. Negosiasikan ruang lingkup dasar
3. Menilai kelayakan proyek dasar
4. Mengembangkan jadwal dasar dan anggaran
5. Mengkomunikasikan rencana proyek

Salah satu tugas terpenting dari fase definisi ruang lingkup adalah membangun dasar awal dari masalah, peluang atau arahan yang memicu proyek. Setiap masalah, kesempatan, dan direktif dinilai sehubungan dengan urgensi, visibilitas, manfaat nyata, dan prioritas. Setiap tambahan, analisis rinci adalah tidak relevan pada tahap proyek ini. Mungkin, bagaimanapun, akan berguna untuk daftar kendala yang dirasakan (batasan) pada proyek, seperti tenggat waktu, anggaran maksimum atau teknologi umum. Seorang analis sistem senior atau manajer proyek biasanya memimpin tugas ini. Sebagian besar peserta lain secara luas diklasifikasikan sebagai pemilik sistem. Ini termasuk eksekutif sponsor, manajer tingkat tertinggi yang akan membayar dan mendukung proyek. Ini juga termasuk manajer dari semua unit organisasi yang dapat dipengaruhi oleh sistem dan mungkin termasuk manajer sistem informasi. Pengguna sistem, perancang sistem dan pembangun sistem biasanya tidak terlibat dalam tugas ini.

2. Analisis permasalahan (*Problem Analysis*)

Fase analisis masalah mempelajari sistem yang ada dan menganalisis temuan untuk memberikan pemahaman yang lebih menyeluruh kepada tim proyek tentang masalah yang memicu proyek. Pada gambar 2.4. memberikan gambaran grafis dari

fase analisis masalah dalam hal blok bangunan sistem informasi . Perhatikan bahwa *user* masih menyertakan sistem tetapi fase ini juga mulai melibatkan pengguna sistem secara aktif. Pengguna sistem adalah ahli materi pelajaran bisnis dalam proyek apa pun. Perluasan yang disengaja dari perspektif pengguna sistem untuk tumpang tindih dengan banyak fase, dengan prinsip “libatkan pengguna sistem”. Tentu saja, manajer proyek dan analisis sistem selalu terlibat dalam semua fase proyek.

Seperti yang ditunjukkan pada gambar 2.4. prasyarat untuk fase analisis masalah adalah ruang lingkup dan pernyataan masalah sebagaimana didefinisikan dan disetujui dalam fase definisi ruang lingkup. Hasil dari fase analisis masalah adalah seperangkat tujuan pembuatan sistem yang berasal dari pemahaman menyeluruh tentang masalah bisnis. Tujuan ini tidak mendefinisikan input, output, atau proses. Sebaliknya, mereka menentukan kriteria bisnis di mana setiap sistem baru akan dievaluasi.

Tujuan dari tahap analisis masalah adalah untuk mempelajari dan memahami domain masalah dengan cukup baik untuk menganalisis masalah, peluang, dan batasannya secara menyeluruh. Beberapa metodologi mendorong pemahaman yang sangat rinci tentang sistem saat ini dan mendokumentasikan sistem itu secara detail menggunakan model sistem seperti diagram aliran data. Hari ini, kecuali ketika proses bisnis harus didesain ulang, upaya yang diperlukan dan nilai tambah oleh pemodelan rinci tersebut dipertanyakan dan biasanya dilewati. Dengan demikian, versi saat ini dari metodologi cepat hipotetis kami hanya mendorong pemodelan sistem yang cukup untuk memperbaiki pemahaman tentang ruang lingkup proyek dan pernyataan masalah, dan untuk mendefinisikan kosakata umum untuk sistem.

Tahap analisis masalah terutama berkaitan dengan pemilik sistem dan pengguna sistem dari sistem yang ada. Daftar dibuat dalam tahap investigasi awal untuk menganalisis pengetahuan, proses, dan blok bangunan komunikasi dari sistem yang ada.

Perhatikan juga bahwa menyiratkan pemodelan sistem minimal. Kami masih dapat menggunakan kerangka potongan untuk menganalisis masing-masing blok bangunan untuk masalah, sebab, dan akibat.

Tahap akhir dari penyampaian dan pencapaian adalah untuk menghasilkan tujuan perbaikan sistem yang mengatasi masalah, peluang, dan arah. Bergantung pada ukuran dan kompleksitas sistem dan seberapa baik kelayakan proyek diketahui, tugas-tugas yang diilustrasikan dapat memakan waktu mulai dari satu hingga enam minggu. Fase analisis masalah biasanya mencakup tugas-tugas berikut:

- a. memahami masalah domain
- b. menganalisis masalah dan peluang
- c. menganalisis proses bisnis
- d. menetapkan tujuan perbaikan sistem
- e. memperbarui atau memperbaiki rencana proyek
- f. mengkomunikasikan temuan dan rekomendasi

3. Analisis kebutuhan (*Requirements Analysis*)

Fase analisis persyaratan mendefinisikan dan memprioritaskan kebutuhan bisnis. Secara sederhana, analisis mendekati pengguna untuk mencari tahu apa yang mereka butuhkan atau inginkan dari sistem baru, dengan hati-hati menghindari diskusi tentang teknologi atau implementasi teknis. Ini mungkin fase yang paling penting dari pengembangan sistem. Kesalahan dan kelalaian dalam analisis persyaratan menghasilkan ketidakpuasan pengguna dengan sistem akhir dan modifikasi yang mahal.

Pada fase ini yang memiliki peran paling besar mencakup pengguna sistem (yang mungkin termasuk pemilik yang akan benar-benar menggunakan sistem) dan analisis sistem dan manajer proyek juga terlibat.

System designers dihilangkan dari fase ini untuk mencegah perhatian dini terhadap solusi teknologi. Blok bangunan itu sendiri dapat menyediakan kerangka kerja untuk mendefinisikan banyak kebutuhan bisnis, termasuk persyaratan data bisnis, persyaratan proses bisnis dan persyaratan antar muka bisnis dan sistem.

Pada fase ini melihat bahwa tujuan peningkatan sistem dari tahap analisis masalah merupakan prasyarat untuk tahap analisis kebutuhan. Kiriman adalah pernyataan kebutuhan bisnis. Sekali lagi, pernyataan persyaratan ini tidak menentukan kemungkinan atau solusi teknis apa pun. Pernyataan persyaratan dapat berupa dokumen sekecil beberapa halaman, atau mungkin luas dengan satu halaman atau lebih dokumentasi untuk setiap persyaratan.

Untuk menghasilkan pernyataan kebutuhan bisnis, analis sistem bekerja sama dengan pengguna sistem untuk mengidentifikasi kebutuhan dan prioritas. Informasi ini dikumpulkan dengan cara wawancara, kuesioner, dan pertemuan yang difasilitasi. Tantangan bagi tim adalah untuk memvalidasi persyaratan tersebut. Tujuan peningkatan sistem menyesuaikan untuk kebutuhan bisnis apakah setiap persyaratan untuk menghasilkan pernyataan kebutuhan bisnis, analis sistem bekerja sama dengan pengguna sistem untuk mengidentifikasi kebutuhan dan prioritas. Informasi ini dikumpulkan dengan cara wawancara, kuesioner, dan pertemuan yang difasilitasi.

Tantangan bagi tim adalah untuk memvalidasi persyaratan tersebut. Tujuan perbaikan sistem menyediakan "kunci penilaian" untuk kebutuhan bisnis. Dengan cara mempertimbangkan kebutuhan pengguna. Biasanya, persyaratan juga harus diprioritaskan[7].

Pada sebuah perancangan sistem informasi terdapat fase perancangan dan analisis sistem dalam suatu sistem informasi terdapat beberapa kebutuhan yang dibutuhkan dalam perancangan sistem informasi serta menganalisis sistem yang ada

digambarkan *use case* diagram, oleh *flowmap*, data *flow* diagram, (DFD) fisik dan *entity relationship* diagram (ERD).

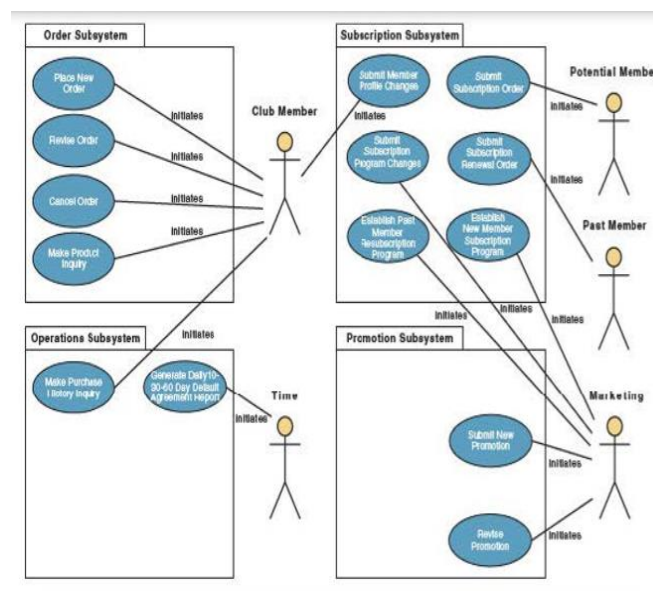
Fase desain sistem umum, membuat *flowmap* usulan, data *flow diagram* (DFD) logic yang menjadi rancangan basis data, rancangan antar muka dari sistem informasi dengan sofwer microsof visio. Pada fase desain sistem umum, membuat *conceptual* data model (CDM), logical data model (LDM) dengan menggunakan *software* power designer dan stuktur data. *conceptual* data model (CDM), *logical* data model (LDM), *physical* data model (PDM) dan struktur data inilah yang menjadi rancangan sistem informasi[8]. Semua kebutuhan dalam sistem informasi yang di jelaskan di atas bertujuan untuk intergrasi yaitu kemampuan untuk memahami berbagai aspek sistem[9]. Dalam pengembangan sistem alat yang umum digunakan antara lain Microsoft Visio 2013, PowerDesigner 16.6, dan Microsoft Access 2013[10]. Metodologi yang umum digunakan untuk pengembangan sistem adalah software life cycle (SDLC), yaitu suatu metode di mana perangkat lunak yang berkualitas dikembangkan sesuai dengan keinginan pembuat dalam jangka waktu tertentu. Langkah-langkah dalam siklus hidup pengembangan sistem dimulai dengan mendefinisikan ide, membuat model data konseptual, membuat logika desain dan desain fisik database berdasarkan database[11].

2.6 . Alat bantu perancangan sistem

2.6.1. Use case diagram

Use Case bertujuan untuk mendeskripsikan atau memodelkan proses dan entitas yang saling terkait dari rangkaian skenario gabungan[12]. Teknik yang digunakan dalam pengembangan sistem atau perangkat lunak untuk mendapatkan persyaratan fungsional dari suatu sistem merupakan pengertian dari diagram *use case*. Diagram ini digunakan untuk membangun hal-hal perilaku pada suatu model, hal tersebut dikarenakan diagram dapat menjabarkan interaksi yang terjadi antara pengguna dan sistem. Selain itu, fungsionalitan dan fitur dari perangkat lunak berdasarkan pandangan pengguna dapat ditentukan dengan menggunakan diagram ini[13].

Use case diagram adalah diagram yang menggambarkan tindakan antara pengguna sistem dengan sistem dan sistem eksternal. Dapat dikatakan secara grafis mengenai pengguna sistem dan hubungan antara jalan penggunaan suatu sistem. Pada *use case diagram* terdapat beberapa komponen fungsional yang mendekomposisi tindakan atau yang memecah sistem menjadi subkomponen[14]. Contoh *use case diagram* dapat dilihat pada Gambar 2.5.



Gambar 2.5 Contoh *Use case diagram*

Adapun komponen pada diagram ini sebagai berikut:

1. *Use case*

Use case adalah situasi di mana serangkaian langkah (skenario) terkait digunakan, baik secara manual atau otomatis, untuk menyelesaikan bisnis. Secara grafis digambarkan sebagai elips horizontal dengan nama shell di bagian atas, bawah, atau di dalam elips. *Use case* merepresentasikan tujuan sistem dalam serangkaian aktivitas dan interaksi pengguna untuk mencapai tujuan tersebut *Use case* adalah teknik yang sangat baik untuk mendokumentasikan

persyaratan sistem dan memahaminya. Skenario yang digunakan dalam kasus penggunaan terdiri dari satu atau lebih persyaratan dan tidak dianggap sebagai persyaratan fungsional. Salah satu simbol *Use case* dapat dilihat pada gambar 2.6.



Gambar 2.6 Simbol *use case*

2. Aktor

Aktor adalah segala sesuatu yang perlu berinteraksi dengan sistem untuk bertukar informasi. Ada empat jenis pelaku yaitu pelaku usaha utama adalah pemangku kepentingan yang diuntungkan dari implementasi use case dengan menerima sesuatu yang dapat diamati atau diukur, nilai label. Orang-orang bisnis ini mungkin atau mungkin tidak menunjukkan aktivitas bisnis. Pelaku sistem utama adalah pemangku kepentingan yang secara langsung mengaitkan antarmuka dengan peristiwa bisnis, pelaku sistem, atau bisnis. Aktor utama adalah orang yang sama dalam peristiwa yang berinteraksi langsung dengan sistem. Pemangku kepentingan yang menanggapi suatu kasus adalah orang yang menanggapi kasus tersebut. Pemangku kepentingan bukanlah pemain kunci, yaitu orang-orang yang mendapatkan nilai yang terukur[14]. Simbol aktor dapat dilihat pada gambar 2.7.

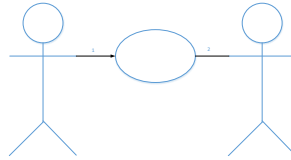


Gambar 2.7 Simbol aktor

3. *Relationship*

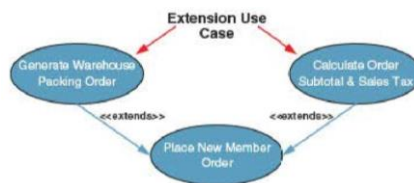
Relationship atau hubungan digambarkan sebagai garis antara dua symbol pada diagram *use case*. Hubungan ini memiliki arti yang berbeda, arti tergantung pada

bagaimana garis terhubungkan dan jenis symbol apa yang dipilih. Asosiasi adalah hubungan antara aktor dan *use case* dimana interaksi terjadi diantara mereka. Contoh *association relationship* dapat dilihat pada gambar 2.8.



Gambar 2.8 Contoh *association relationship*

Extension use case adalah diekstraksi antara *use case* yang lebih kompleks di order untuk menyederhanakan kasus asli sehingga dungsihnya dapat diperluas. Adapun contoh dari *extension relationship* dapat dilihat pada Gambar 2.9.



Gambar 2.9 Contoh *extension relationship*

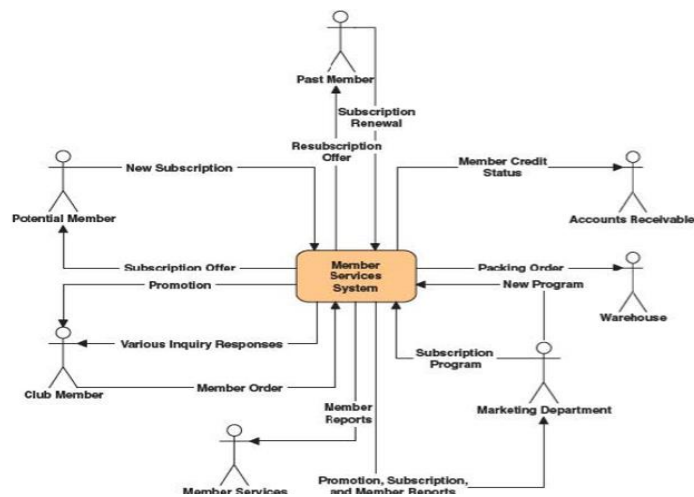
2.6.1 Context diagram

Diagram konteks sangat membantu dalam kasus yang penggunaannya berbeda[7]. Diagram konteks yaitu diagram dengan alur data tingkat konteks. Konteks sendiri memiliki arti yakni kondisi terjadinya suatu keadaan. Permodelan konteks dapat disebut sebagai representasi konteks. Semakin banyak konteks yang ditampilkan pada diagram maka semakin sedikit deskripsi tekstual yang dibutuhkan. Diagram ini digunakan dalam membuat diagram serta mendokumentasikan sistem perangkat lunak, sehingga pembuat diagram ini dapat melihat gambaran besar. Orang seperti aktor, peran, dan sistem perangkat lunak dari teknologi, protokol dan detail lainnya menjadi fokus dari diagram[7].

Tujuan diagram konteks yaitu untuk menganalisis bagaimana sistem berinteraksi dengan dunia serta menentukan *output* serta *input* secara umum dari sistem[7]. Diagram ini mempatkan sistem dalam konteks lingkungan, yang terdiri dari simbol

berjumlah satu yang menggambarkan seluruh sistem. Adapun ketentuan dalam menggambar diagram konteks, yakni:

1. Menggunakan hanya satu simbol proses.
2. Label simbol proses digunakan untuk menggambarkan pada sistem secara keseluruhan. Dapat pula ditambahkan keterangan objek.
3. Simbol proses tidak diberi nomor.
4. Terminator dari sistem harus disertakan.
5. Arus data antara sistem dan terminator harus dapat ditunjukkan.



Gambar 2.10 Contoh konteks diagram

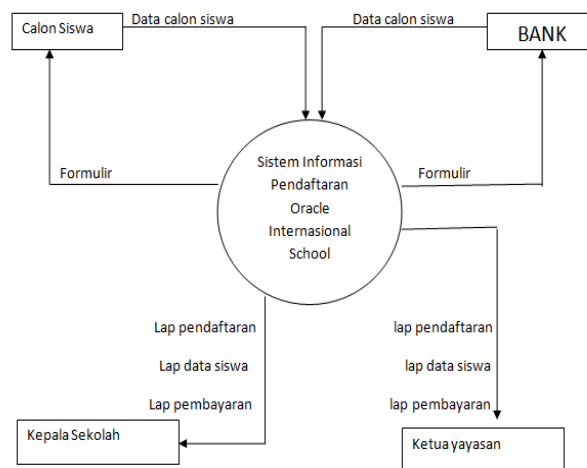
Adapun simbol yang digunakan dalam diagram konteks, yaitu:

Notasi Yourdan/Demarcó	Notasi Gane & Sarson	Keterangan
		Simbol <i>external entity</i> / Terminator menggambarkan asal atau tujuan data di luar sistem
		Simbol lingkaran menggambarkan entitas atau proses dimana aliran data masuk ditransformasikan ke aliran data keluar
		Simbol aliran data menggambarkan aliran data
		Simbol file menggambarkan tempat data disimpan

Gambar 2.11 Simbol diagram konteks

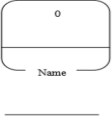
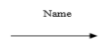
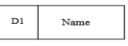

2.6.2 Data flow diagram (DFD)

Diagram aliran data (data *flow* diagram) adalah representasi grafis dari suatu sistem yang menggunakan berbagai bentuk simbolik untuk menggambarkan data yang mengalir melalui proses terkait. Cara paling alami untuk mendokumentasikan proses ini adalah dengan menggunakan DFD[7]. Diagram aliran data digambarkan sebagai pergerakan data antara entitas prosedural, eksternal, dan penyimpanan data dalam sistem. Diagram juga dapat dikatakan sebagai alat untuk menyatakan hubungan antara berbagai komponen dalam suatu program atau sistem. Tao dan Kung menunjukkan bahwa teknik grafik aliran data efektif untuk menggambarkan kebutuhan fungsional dari sistem besar yang kompleks[14].



Gambar 2.12 Contoh DFD

Teknik dalam diagram sangat efektif dalam mengekspresikan kebutuhan fungsional dari sistem yang kompleks. Simbol pada diagram adalah alat grafis yang berbeda dan bermakna, sehingga dapat membantu mengatasi ketidak konsistenan dan kebingungan. Ada empat simbol aliran dalam diagram, yaitu aliran data, proses, penyimpanan data, dan entitas eksternal. Secara umum, ada dua gaya yang sering digunakan.[14]. Adapun simbol DFD, sebagai berikut:

Symbol	Element Name
	Process
	Data Flow
	Data Store
	External Entity

Gambar 2.13 Simbol DFD

Tampilan tertinggi pada diagram disebut dengan diagram konteks. Level selanjutnya disebut dengan diagram aliran data level 0 mewakili proses utama pada aliran data, proses utama sistem, dan penyimpadan data yang sangat detail. Proses diagram aliran data level n-1 didekomposisikan menjadi diagram aliran data level bawah yang merupakan diagram aliran data level n. Diagram ini memiliki prinsip utama yaitu untuk memastikan keseimbangan aliran data pada satu tingkat dengan akurat yang diwakilkan dalam diagram aliran data tingkat berikutnya dalam megembangkan proyek[14]. Langkah-langkah untuk membuat diagram arus data agar meningkatkan efektivitas, sebagai berikut:

1. Tandai aliran data dengan nama unik. Nama yang unik akan memudahkan untuk menjelaskan setiap aliran data dalam data.
2. Nama aliran data konsisten dari setiap lapisan
Kontinuitas terjadi dalam aliran data dari satu level ke level lainnya. Bila namanya sudah konsisten maka bisa dikatakan DFD seimbang (*balanced DFD*).
3. Tunjukkan lokasi yang benar
Untuk catatan yang dihapus dari penyimpanan data, perlu untuk menunjukkan bahwa mereka mengantisipasi kebutuhan untuk pemulihan data.
4. Secara opsional, sertakan proses baca dan tulis

Jangan sertakan prosedur membaca dan menulis saat merekam program komputer. Perlu disertakan adalah langkah-langkah untuk mengubah data.

5. Hindari proses membaca saja

Ketika sistem hanya memiliki proses data masuk tanpa data keluar, ada kesalahan yang terjadi. Maka proses tersebut diberi nama seperti *black hole*.

6. Mengizinkan proses membaca

Proses membaca diizinkan ketika waktu berfungsi sebagai pemicu. Tindakan atau berlalunya waktu dapat digunakan sebagai awal proses. Penerimaan dokumen adalah contoh dari tindakan yang dilakukan[15].

2.6.3 Entity relationship diagram (ERD)

Diagram hubungan entitas dikenal pula sebagai diagram chen karena penyempurnaan dari model Song & Chissom dalam penggunaannya untuk memprediksi jumlah pendaftaran[16]. Metode chen digunakan untuk menggambarkan hubungan antara model dan sistem[17]. Diagram hubungan entitas (*entity relationship diagram*) mendokumentasikan perusahaan dengan mengidentifikasi jenis entitas beserta dengan hubungannya. ERD adalah alat pembuatan model yang paling fleksibel dapat digunakan sebagai pendekatan pengembangan perusahaan. Titik disiapkan untuk ERD sebagai proses pengembangan sistem saat “gambaran besar” data telah ditentukan. Titik tersebut hadir saat:

1. Saat keseluruhan model data dilibatkan pada eksekutif perusahaan dengan memperhatikan berbagai kebutuhan pada perusahaan.
2. Saat pembuatan model segmen operasi perusahaan yang besar melibatkan eksekutif.

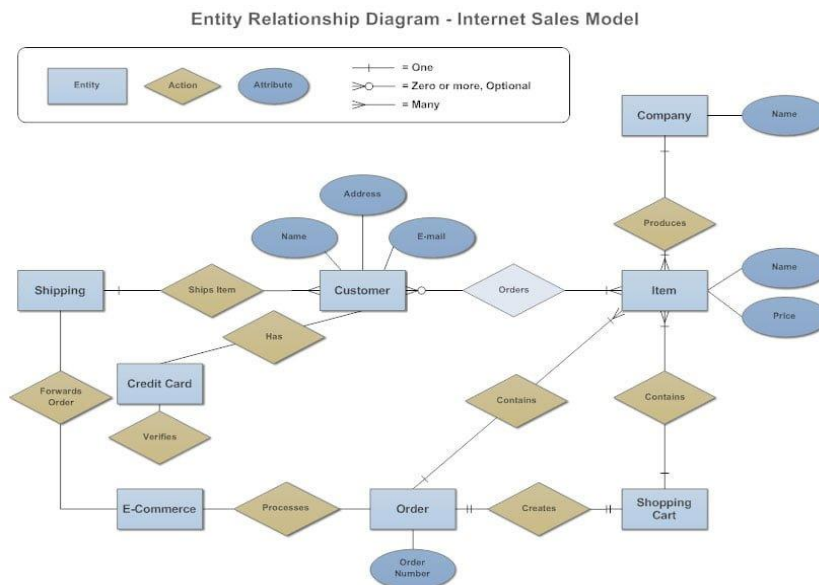
3. Saat pembuatan model data untuk penerapan sesuatu melibatkan sesialis informasi.

Langkah penting dalam rekayasa perangkat lunak dan desain sistem informasi adalah pembuatan diagram yang solid. *Entity-relationship* diagram adalah teknologi utama dari diagram, yaitu untuk merekam entitas dan jenis hubungan dalam bentuk grafik. Simbol-simbol dalam diagram ini telah berevolusi. Beberapa notasi menggambarkan properti yang dimodelkan pada hubungan. Fitur grafik, termasuk:

1. Tipe objek diwakili oleh entitas yang dapat membedakan keduanya. Tipe entitas adalah objek bisnis penting yang berisi lebih dari satu properti. Entitas yang lemah adalah tipe khusus dari entitas dimana keberadaannya bergantung dengan entitas lain atau bisa disebut entitas pemilik.
2. Tipe hubungan mewakili hubungan antara atau di antara beberapa entitas. Dalam pemodelan dunia nyata, suatu hubungan mewakili suatu asosiasi yang perlu diingat oleh sistem basis data. Kami hanya akan memanggil hubungan, bukan tipe hubungan.
3. Atribut merupakan properti untuk menggambarkan hubungan atau entitas. Beberapa metode dapat membuat atribut berada dalam satu hubungan. Atribut yang merupakan *primary key* dari relasi lain yang dapat disebut *foreign key*.
4. Jumlah instansi hubungan tempat entitas ditentukan oleh batasan kardinalitas. Bentuk dari batasan tersebut yaitu 1:1, 1:N, M:N, dalam hubungan biner, dan 1:1:2, 1:1:N, 1:N:M, dan M:N:P dalam hubungan ternary. Batasan disesuaikan dengan kardinalitas maksimum dalam beberapa notasi.
5. Batasan partisipasi menentukan mengenai hubungan instansi entitas dapat ada tanpa berkaitan dengan entitas lain. Batasan sesuai dengan batasan minimum pada notasi. Partisipasi terbagi menjadi dua yaitu partisipasi total (wajib) dan parsial (opsional). Partisipasi total yaitu ketika instansi entitas dinyatakan tidak

ada jika instansi entitas tidak berhubungan dengan instansi entitas lainnya. Partisipasi parsial yaitu saat instasi entitas dapat ada tanpa terlibat hubungan dengan instansi entitas yang lain. Beberapa metode menggabungkan kardinalitas dan batasan partisipasi dan mewakilinya dengan batasan minimum dan maksimum dalam bentuk notasi dalam penggunaanya.

6. Generalisasi penetapan hubungan *superclass* dan *subclass* antar tiap tipe entitas. Ada dua poin kendala jika ditinjau pada hirarki generalisasi yaitu *disjoint* dan *complete*. Batasan *disjoint* digunakan untuk menentukan entitas dapat muncul pada lebih dari satu *subclass* (tumpang tindih) atau tidak (*disjoint*). Spesialisasi memungkinkan tumpang tindih apabila satu instansi entitas pada *superclass* tidak dapat muncul di entitas *subclass*. Jika tidak, subclass terpisah. Kendala yang kedua yaitu mengenai kelengkapan. Kelengkapan untuk menentukan apakah instansi entitas *superclass* akan ada walaupun tidak memiliki satu entitas *subclass* (spesialisasi sebagian) atau tidak (total spesialisasi)[18] .



Gambar 2.14 Contoh ERD

Diagram hubungan entitas terdapat komponen pembentuknya, yaitu:

1. Jenis entitas

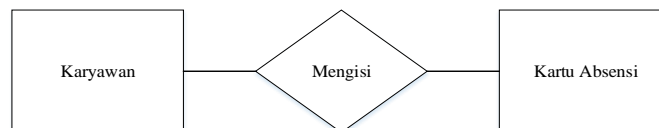
Jenis entitas dapat berupa sumber daya, transaksi atau elemen lingkungan, yang penting bagi perusahaan sehingga didokumentasikan dengan data. Contoh dari entitas adalah pelanggan, kartu absensi. Jenis entitas didokumentasikan pada diagram dengan bentuk persegi empat yang diberi label nama jenis entitas, kerap pemberian nama berupa kata tunggal. Entitas ini dapat muncul beberapa kali. Adapun identitas jenis entitas dapat dilihat dibawah ini.



Gambar 2.15 Jenis entitas

2. Hubungan

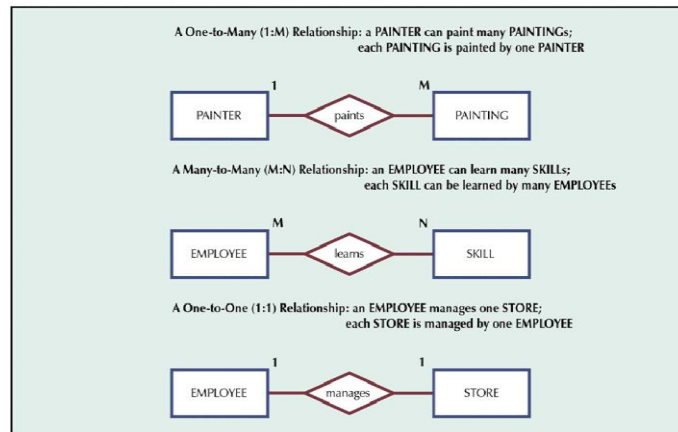
Hubungan atau *relationship* merupakan asosiasi antara dua jenis entitas. Hubungan digambarkan dengan bentuk belah ketupat yang diberi label kerja. Seperti mengisi kartu. Adapun contoh hubungan, yaitu:



Gambar 2.16 Hubungan

3. Keterkaitan

Entitas berhubungan dengan banyak entitas lain disebut sebagai keterkaitan. Jenis keterkaitan itu ada satu- ke-banyak, satu-ke-satu, dan banyak ke banyak. Karakter yang digunakan untuk mengabarkan keterkaitan yaitu karakter 1 dan M. Adapun contoh dari berbagai keterkaitan, sebagai berikut.



Gambar 2.17 Contoh keterkaitan ERD

ERD disiapkan oleh spesialis informasi bekerja sama dengan pemakai. Para pemakai dapat meliputi eksekutif, komite pengarah SIM, manajer area pemakai, serta non-manajer. Dalam pembuatan *entity relationship diagram*, terdapat beberapa langkah sebagai berikut:

1. Mengidentifikasi entitas.

Manajemen menentukan elemen lingkungan, sumber daya dan transaksi yang mana akan dijelaskan dengan data.

2. Mengidentifikasi hubungan

Setiap entitas dihubungkan dengan entitas lain melalui suatu tindakan.

3. Menyiapkan ERD kasar

Simbol dibuat sketsa sehingga symbol dapat terbaca dari kiri ke kanan.

4. Memetakan elemen elemen pada entitas

Elemen elemen data yang mengidentifikasi dan menjelaskan setiap entitas pada data didaftarkan di sebelah entitasnya.

5. Membuat analisis data

Elemen-elemen data dipelajari untuk membuat struktur database menjadi efisien. Proses melaksanakan analisis data disebut normalisasi, dan tugasnya adalah menyesuaikan data sehingga serupa dengan serangkaian berbentuk normal.

6. Menyiapkan ERD yang telah dimodifikasi

Hasil dari analisis data disatukan ke dalam suatu ERD baru. Dengan demikian jenis entitas dan hubungannya diatur sehingga memberikan dasar yang paling efisien untuk database.

7. Menelaah ERD bersama pemakai dan memperbaikinya

Spesialis informasi menelaah diagram bersama eksekutif, manajer dan non manajer pada area pemakai dan diperbaiki jika perlu.

- a. Pada *navigation pane*, pilih nama objek tabel yang akan digunakan untuk sumber data *form*.

Klik *tab create* dan klik tombol *form* pada *group forms*, dengan demikian akan terlihat bentuk *form* berisikan *field* dengan urutan berdasarkan sumber data.