Microsoft Solution Framework sebagai Model Proses Pengembangan Perangkat Lunak Berbasis *Milestone*, Tinjauan pada Fase Envisioning dan Planning

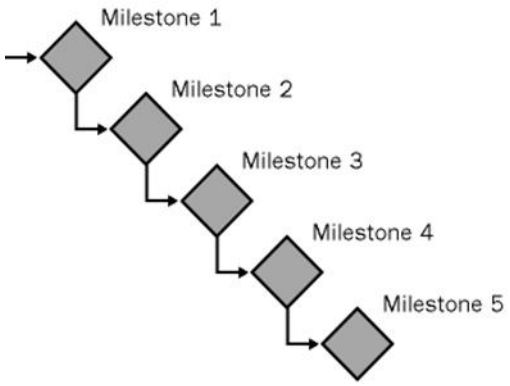
Migunani
Program Studi Sistem Informasi STMIK ProVisi Semarang
email : -

Abstrak : Prinsip-prinsip metodologi perancangan perangkat lunak menjadi basis proses rekayasa aplikasi bisnis telah banyak di terapkan. Dimana sebuah metodologi merupakan kerangka pijakan utama dalam perancangan perangkat lunak profesional untuk menghasilkan aplikasi yang sesuai dengan kebutuhan bisnis sebuah organisasi. Tahapan-tahapan dalam membangun sebuah perangkat lunak akan sangat berguna untuk memastikan apakah elemen-elemen proyek pengembangan perangkat lunak telah dikelola dengan baik dan benar. Microsoft Solution Framework (MSF) sebagai metodologi alternatif perancangan perangkat lunak menggabungkan dua metodologi yang berbeda menjadi satu kesatuan yang utuh untuk menghasilkan sebuah solusi perangkat lunak yang lebih dinamis dengan mengadopsi kelebihan dari masing-masing metodologi. Kedua metodologi tersebut adalah Waterfall dan Spiral. Metodologi Waterfall menggambarkan sebuah model proses yang statis dengan tahapan-tahapan berlapis yang menggunakan sebuah milestone sebagai transisi pada setiap tahap perancangan, sementara metodologi Spiral (iterative) menerapkan model proses secara sirkular tanpa adanya cekpoint atau milestone. Namun kelebihan metodologi spiral adalah mengenai kebutuhan pengembangan secara keberlanjutan dan adanya keterlibatan pemakai dalam pembangunan perangkat lunak sehingga perangkat lunak akan selalu berkembang dengan versi dan fitur yang lebih baru.

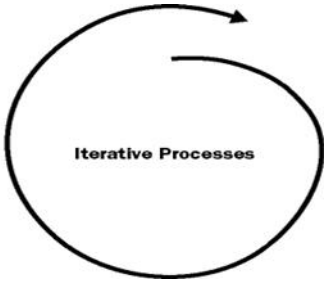
Kata kunci : metodologi, perangkat lunak, microsoft slution famework.

PENDAHULUAN

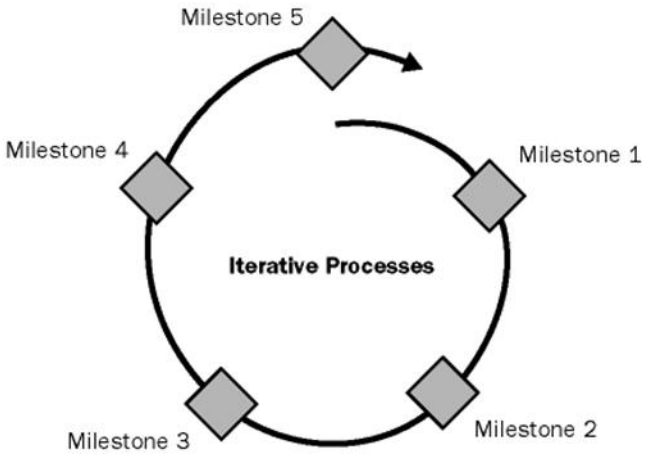
Keberhasilan pengembangan perangkat lunak bergantung pada pengelolaan proyek perangkat lunak secara keseluruhan. Menetapkan sebuah metodologi yang memiliki dinamisasi tinggi dalam tahap-tahap perancangan model proses perangkat lunak akan sangat berpengaruh pada kualitas perangkat lunak yang dihasilkan. Model proses merupakan tahap-tahap aktivitas proyek yang menggambarkan daur hidup suatu proyek. *Microsoft Solution Framework (MSF)* adalah metodologi perancangan dan pengembangan aplikasi bisnis yang diperkenalkan oleh sebuah vendor *software* besar yaitu *Microsoft Corporation*. Prinsip-prinsip pengembangan perangkat lunak dengan metodologi *MSF* memiliki perencanaan berbasis milestone (model waterfall), dan memberikan hasil yang dapat diprediksi (model spiral/iterative) disertai umpan balik dan kreativitas dari tim pengembang.



Gambar 1. Model Waterfall



Gambar 2. Model Spiral/Iterative



Gambar 3. Model Microsoft Solution Framework (MSF)

1. Model Proses Waterfall (S*equential*)

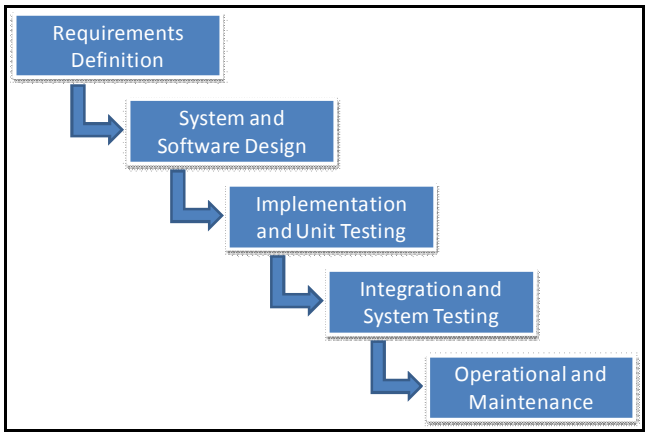
Model ini menggunakan milestone sebagai titik transisi dan pengujian, artinya setiap aktivitas pada tahap pengembangan harus diselesaikan sebelum menuju tahap pengembangan berikutnya. Sehingga model ini sangat sesuai untuk perangkat lunak dengan syarat-syarat yang telah didefinisikan secara lengkap sebelumnya karena besar kemungkinan tidak adanya perubahan aplikasi dimasa yang akan datang. Kondisi semacam ini akan sangat berpengaruh pada perangkat lunak dan menimbulkan masalah terhadap kebutuhan iterasi dimana aplikasi akan terus berkembang dengan penyesuaian-penyesuaian terhadap kebutuhan, proses bisnis dan lingkungan aplikasi yang terus berubah dari waktu kewaktu. Namun kelebihan dari model ini adalah karena adanya titik transisi yang jelas pada setiap tahap, maka akan memudahkan tim pengembang perangkat lunak dalam memonitor penjadwalan proyek, menetapkan tanggung jawab dan akuntabilitas peran personal dalam proyek perangkat lunak.

Kelemahan dari model ini adalah adanya kendala dalam mengakomodasi perubahan setelah proses pengembangan telah berjalan. Fase sebelumnya harus lengkap dan selesai sebelum memasuki tahap berikutnya. Beberapa kendala yang muncul pada model waterfall adalah :

a. Aspek perubahan pada perangkat lunak sulit dilakukan karena sifatnya yang kaku dimana kebutuhan perangkat lunak harus lengkap.

b. Karena sifat kakunya, model ini cocok manakala kebutuhan telah dikumpulkan secara lengkap sehingga perubahan bisa ditekan sekecil mungkin. Akan tetapi pada kenyataannya sangat jarang sekali pengguna dapat mendefinisikan kebutuhan awal secara lengkap, oleh karena perubahan kebutuhan adalah sesuatu yang wajar terjadi.

c. Waterfall pada umumnya digunakan untuk rekayasa sistem perangkat lunak berkapasitas besar dimana proyek dikerjakan di beberapa tempat berlainan, dan terbagi menjadi beberapa bagian sub-proyek.

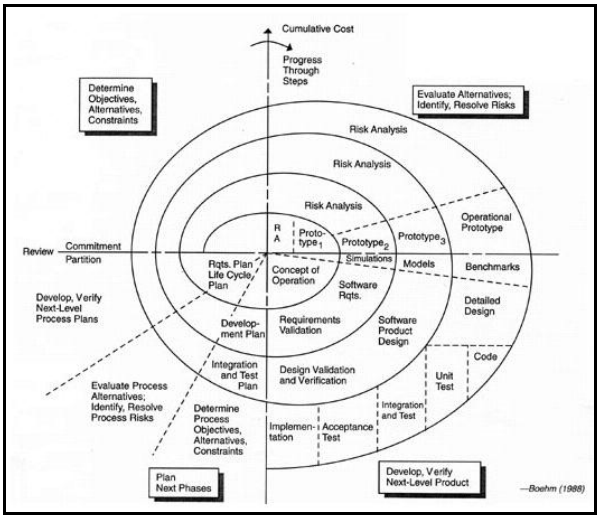


Gambar 3. Tahapan Pada Model Proses Waterfall (Sommervile, 2001)

2. Model Proses Spiral (*Iterative*).

Model ini berbasiskan pada kebutuhan keberlanjutan untuk menyaring kebutuhan-kebutuhan tersebut dan estimasi proyek secara keseluruhan. Model ini menerapkan perancangan model proses yang lebih dinamis dengan terus beradaptasi terhadap kebutuhan proses bisnis dimasa yang akan datang sehingga aplikasi memiliki versi yang berbeda dengan fitur-fitur yang mengalami peningkatan dari waktu kewaktu.

Kebutuhan waktu untuk pengembangan aplikasi yang cepat dengan kapasitas proyek yang relatif kecil sangat relefan dengan model spiral ini. Keterlibatan pelanggan dengan tim pengembang perangkat lunak akan sangat sering terjadi karena pelanggan akan memberikan *feedback* dan persetujuan setiap tahap dalam pengembangan aplikasi perangkat lunak. Dengan adanya *feedback* dari pelanggan maka estimasi waktu terhadap penyelesaian proyek perangkat lunak menjadi semakin jelas.



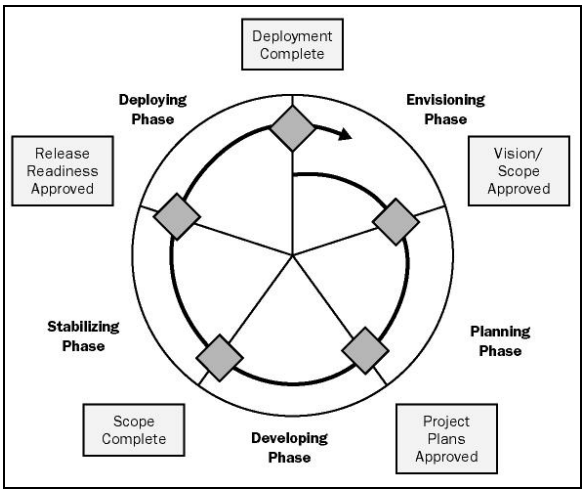
Gambar 4. Tahapan Pada Model Proses Spiral (Boehm, 1988)

Pengembangan perangkat lunak dengan model spiral memiliki kelemahan karena tidak adanya *milestone* sebagai titik transisi dan pengujian maka dikhawatirkan proses pengembangan sistem akan mengalami kekacauan dari segi waktu penyelesaian solusi sistem. Oleh karenanya model ini hanya sesuai untuk aplikasi-aplikasi kecil yang tidak terintegrasi dan terdistribusi.

3. Model Proses Microsoft Solution Framework (MSF).

Model ini menggabungkan dua macam model proses terdahulu dengan menerapkan prinsip-prinsip terbaik dari masing-masing model yaitu pada model *waterfall* dimana titik transisi yang jelas pada setiap tahap pengembangan sistem, maka akan memudahkan tim pengembang perangkat lunak dalam memonitor penjadwalan proyek, menetapkan tanggung jawab dan akuntabilitas peran personal dalam proyek perangkat lunak, sedangkan pada model spiral berbasiskan pada kebutuhan keberlanjutan untuk menyaring kebutuhan-kebutuhan sistem dan estimasi proyek secara keseluruhan. Selain itu model ini menerapkan perancangan model proses yang lebih dinamis dengan terus beradaptasi terhadap kebutuhan proses bisnis dimasa yang akan datang sehingga aplikasi memiliki versi yang berbeda dengan fitur-fitur yang mengalami peningkatan dari waktu kewaktu.

Model proses *Microsoft Solution Framework* akan menguraikan urutan aktivitas yang di generalisasi untuk membangun dan menyebarkan solusi perusahaan (*enterprise*), fleksibel dan dapat mengakomodasi pengembangan dan desain yang luas, berbasiskan fase (tahapan) seperti pada model waterfall, dan dapat diterapkan untuk pengembangan perangkat lunak yang yang memiliki kontinuitas dengan menyesuaikan kebutuhan aplikasi bisnis dengan versi-versi yang lebih baru.



Gambar 5. Tahapan Pada Model Proses MSF. (Microsoft, 2003)

PENGEMBANGAN SISTEM PERANGKAT LUNAK DENGAN MODEL PROSES MSF.

Model proses MSF memiliki tahapan-tahapan yang sedikit berbeda dengan model proses waterfall dan spiral. MSF mengenal lima tahapan yang disebut dengan *Envisioning Phase,*

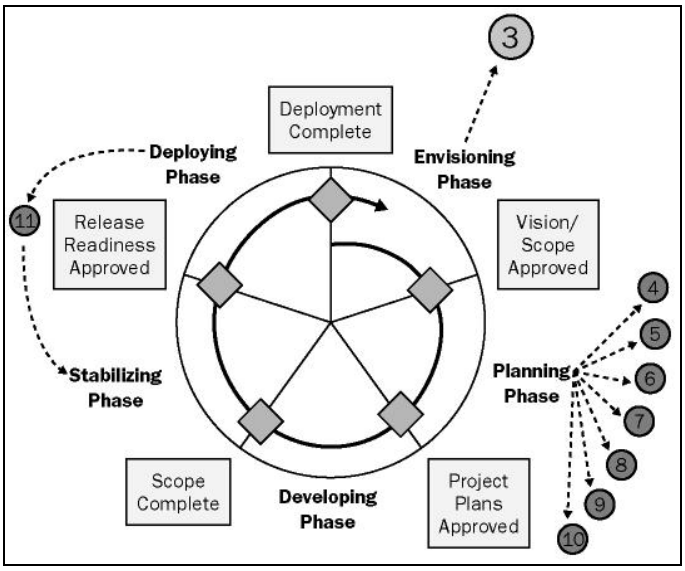
Planning Phase, Developing Phase, Stabilizing Phase, dan *Deploying Phase*. Pada setiap phase merupakan tahapan-tahapan yang berbeda dan mencapai kulminasi pada sebuah *milestone*.

1. *Envisioning Phase.*

Fase ini memberikan gambaran yang lengkap mengenai sistem yang akan dikembangkan, dimana tim pengembang melakukan peninjauan secara umum permasalahan-permasalahan bisnis yang akan di pecahkan dan bagaimana permasalahan tersebut berhubungan dengan bisnis perusahaan, pengguna atau konsumen serta lingkungan perusahaan. Langkah ini akan membantu tim pengembang dalam memperoleh jangkauan yang jelas bagaimana membangun solusi yang tepat untuk menyelesaikan masalah bisnis yang sedang dihadapi.

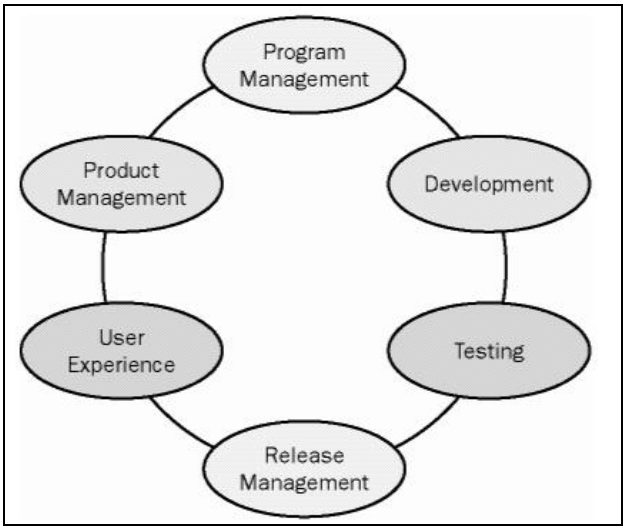
Sebagai contoh apabila sebuah organisasi akan mengembangkan sebuah *website* perusahaan, namun organisasi belum mendapatkan gambaran yang jelas mengenai target jangka panjang yang berhubungan dengan proyek website yang akan dikembangkan, maka pada fase ini target tersebut harus dapat didefinisikan dengan jelas. Untuk merancang sebuah solusi aplikasi yang efektif maka tim harus mengidentifikasi tujuan dari organisasi dengan adanya rencana pengembangan aplikasi website. Selain itu tim juga harus menetapkan karakteristik yang unik untuk *website* yang akan dikembangkan, sehingga organisasi akan memperoleh manfaat yang maksimal. Apabila organisasi menghendaki konten website yang lebih kompleks dan memiliki pengguna yang terus bertambah setiap waktu maka perlu direncanakan teknis yang berhubungan dengan kompleksitas dan skalabilitas dari solusi. Pertimbangan lainya adalah apa yang berhubungan dengan citra perusahaan. Sehingga pada fase ini akan bermanfaat untuk :

a. Mengidentifikasi tujuan dan batasan proyek pengembangan sistem.
b. Menjawab pertanyaan-pertanyaan yang mungkin muncul, memperoleh persetujuan dari *stakeholder* kunci, dan apa saja harapan dari personel yang terlibat didalam proyek pengembangan sistem.
c. Menjadikan basis bagi personel didalam tim proyek dalam pengembangan solusi yang akan datang.
d. Mendefinisikan jangkauan dari proyek, yang mana dapat membantu dalam rencana detail pada fase berikutnya.
e. Memperkirakan sumberdaya yang diperlukan untuk mengembangkan solusi sistem.
f. Melakukan identifikasi dan penjadwalan *milestone* utama yaitu *scope approved*.



Gambar 6. Fase *Envisioning* Model Proses MSF.

Dalam proyek MSF semua personal yang terlibat akan bekerja sebagai unit tunggal. Meskipun demikian masing-masing personel memiliki peran yang berbeda dan fokus pada peran masing-masing sebagai anggota tim pada setiap fase yang akan dilalui.



Gambar 7. Peran (Role) Pada Model Proses MSF.

a. *Product Management* : memiliki tanggung jawab yang berhubungan dengan manajemen produk perangkat lunak, termasuk juga memastikan kebutuhan dari pengguna. *product management* berkolaborasi dengan *program management* berupaya untuk menetapkan jangkauan dari proyek. Dalam menyelesaikan tujuan ini *product management* mempelajari dan menganalisa masalah-masalah bisnis yang terjadi, kebutuhan-kebutuhan bisnis, jangkauan proyek, tujuan bisnis dan profile pengguna.

b. *Program Management* : memiiki tanggung jawab dalam hal memastikan tujuan dari desain proyek, mendefinisikan faktor-faktor keberhasilan proyek, mendefinisikan dengan jelas konsep solusi yang akan dikembangkan, dan mengelola infrastruktur proyek.

c. *Development* : memberikan umpan balik kepada tim pengembang perihal implikasi teknis dari pengembangan perangkat lunak dan kemungkinan-kemungkinan pemilihan konsep solusi yang akan digunakan.

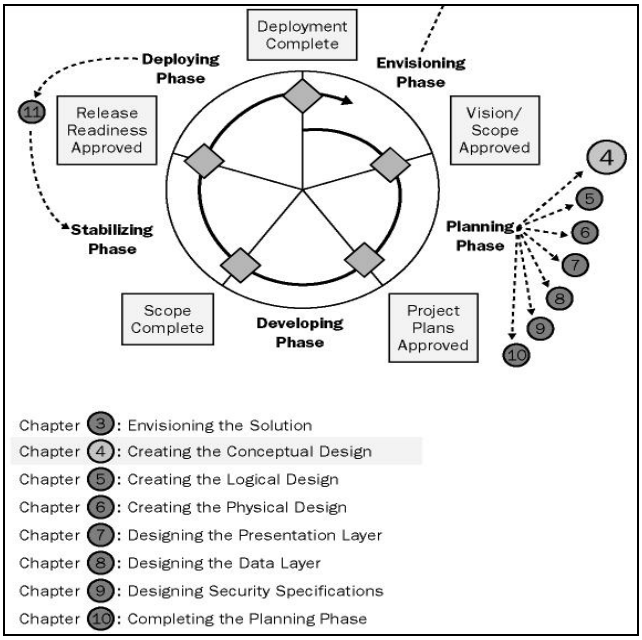
d. *User Experience* : menganalisa kebutuhan performansi terhadap sistem dan dukungan terhadap user dengan mempertimbangkan implikasi dari perangkat lunak untuk mempertemukan kebutuhanya.

e. *Testing* : memberikan umpan balik kepada tim mengenai kualitas solusi dan aktifitas yang dilakukan untuk memperoleh kualitas perangkat lunak.

f. *Release Management* : mengidentifikasi apa yang dibutuhkan apa yang diperlukan untuk menyebarkan solusi, bagaimana dan kapan solusi akan disebarkan, serta infrastruktur tambahan yang diperlukan.

2. *Planning Phase.*

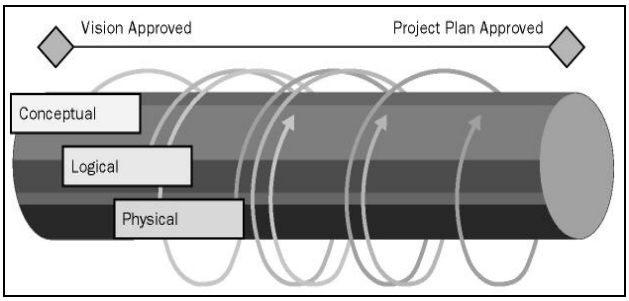
Fase ini akan menjelaskan bagaimana mendefinisikan solusi dengan menetapkan *what* (apa), *how* (bagaimana), dan *who* (siapa) yang berkaitan erat dengan solusi yang akan dibangun. Aktifitas-aktifitas pada fase ini secara lebih detil digambarkan sebagai berikut :



Gambar 8. Fase *Planning* Model Proses MSF.

Pada fase ini akan terbagi menjadi beberapa aktivitas diantaranya adalah membuat desain konseptual, desain logikal, desain fisikal, mendesain lapisan presentasi, mendesain lapisan data dan mendesain spesifikasi tingkat keamanan solusi. Desain konseptual, logikal dan fisikal tidak berjalan secara paralel akan tetapi ketiganya

berlangsung secara bergiliran dan masing-masing tidak saling bergantung satu sama lain.



Gambar 9. Desain Pada Fase *Planning* Model Proses MSF.

a. Desain konseptual.

Melakukan sintesis terhadap informasi yang diperoleh yang menghasilkan model informasi seperti hubungan antara proses bisnis, sistem bisnis, para pemakai, aliran proses dan urutan aktivitas. Selain itu profile pemakai yang telah terbaharukan, kandidat kebutuhan-kebutuhan (*requirements*) dan *usecase* yang lebih detail. Setelah melakukan sintesis terhadap informasi, aktivitas selanjutnya adalah membuat *usage scenario*. Kebutuhan-kebutuhan disini akan dibagi menjadi empat macam diantaranya adalah kebutuhan pemakai, kebutuhan sistem, kebutuhan operasi dan kebutuhan bisnis.

Tabel 1. Draf *Requirement* pada Fase *Envisioning.*

Contoh Draft Statemen *Requirement* pada fase *Envisioning*.	
Req ID	**Requirement**
1	Identifikasi pelanggan berdasarkan produk dan lokasi.
2	Identifikasi penurunan penjualan pada pelanggan.
3	Identifikasi pelanggan potensial.
4	Identifikasi pembeli terbanyak.

Tabel 2. Daftar *Restate Requirement* pada Fase *Planning*

Contoh *Restate Requirement* pada fase *Planning*.	
Req ID	**Requirement**
1.1	Harus dapat menganalisa data pelanggan.
1.1.1	Harus dapat menganalisa keuntungan berdasarkan produk.
1.1.2	Harus dapat menganalisa keuntungan berdasarkan pelanggan.
1.1.3	Harus dapat menganalisa keuntungan berdasarkan daerah.
1.2	Harus dapat mengurutkan (descending, ascending) pelanggan.
1.4	Harus dapat mengidentifikasi *trend* penjualan.
1.4.1	Harus dapat mengidentifikasi turunya penjualan.

Sedangkan *usecase* menggambarkan bagian dari fungsionalitas sistem yang didalamnya terdapat *usage scenario* (sekenario bagaimana *usecase* tersebut difungsikan atau digunakan dengan melibatkan seorang aktor). Sebuah *usecase* dengan nama *Manage Order* dapat memiliki *subordinat usecase Print Order, Create Order, Cancel Order* dan *Revise Order*. *Usecase* tersebut melibatkan dua aktor untuk menjalankan fungsionalitasnya yaitu aktor *Customer* dan *Sales Person*. Aktivitas yang dilakukan untuk memecah *usecase* menjadi *subordinat-subordinat usecase* disebut dengan *refining usecase* (menyuling *usecase*).

Bagaimana seorang aktor melakukan fungsionalitas sistem dapat digambarkan didalam sebuah *usecase* atau *subordinat usecase* yangmana aktivitas-aktivitas tersebut dituangkan didalam sebuah *usage scenario*.

- *Usage Scenario Usecase "Memesan Spesifikasi Produk".*

Tabel 3. *Usage Scenario* dari *Usecase*.

Use case title Spesifikasi Produk	: Memesan
Abbreviated title Spesifikasi Produk	: Memesan
Use case ID	: UC 05.1
Requirements ID	: 15.1
Intent	: Untuk menyediakan spesifikasi produk lengkap kepada pelanggan.
Scenario narrative	: Pada saat pelanggan melihat item dalam katalog, pelanggan meminta spesifikasi-spesifikasi produk.
Actors	: Pelanggan
Preconditions	: Pelanggan membuka katalog produk. Pelanggan melihat item didalam katalog .
Basic course	: 1. *Usecase* dimulai ketika pelanggan meng-klik Pemesanan Spesifikasi Produk untuk item yang dipilih. 2. Pelanggan memilih format dari spesifikasi produk. 3. Pelanggan memilih cara pengiriman produk. 4. Alamat pelanggan telah dikonfirmasi. 5. *Usecase* berakhir ketika pemesanan telah selesai, terkirim dan siap untuk menunggu kiriman katalog.
Alternative course	: -
Postconditions	: Pemesanan telah selesai, terkirim dan siap untuk menunggu kiriman katalog. Pelanggan kembali pada katalog yang dibuka terakhir kali.
Uses/extends	: Pelanggan membaharui profile, pelanggan membuat profile baru, dan mengirimkan pemesanan.
User implementation requests	: Tidak ada
Frequency	: Terjadi antara 17-100 sesi pelanggan di website.
Unresolved issues	: Tidak ada
Authority	: Mike Danseglio
Modification history	: 6 November 2006
Author	: Heidi Steen
Description	: Versi Inisial

- *Usage Scenario Subordinat Usecase "Memesan Spesifikasi Produk Melalui Surat ".*

Tabel 4. *Usage Scenario* dari *Subordinat Usecase*.

Use case title	: Memesan Spesifikasi Produk Melalui Surat
Abbreviated title	: Spesifikasi Produk Melalui Surat
Use case ID	: UC 05.1.1
Requirements ID	: 15.1.1
Intent	: Untuk menyediakan spesifikasi produk yang dipilih kepada pelanggan melalui

Use case title	: Memesan Spesifikasi Produk Melalui Surat
	surat.
Scenario narrative	: Pelanggan memohon spesifikasi produk melalui layanan pengiriman surat. Pelanggan melihat-lihat katalog (UC 05.1)dan melihat item yang diminati, setelah memilih cara pengiriman melalui layanan pos surat, pelanggan mengisikan alamat pengiriman …
Actors	: Pelanggan
Preconditions	: Pelanggan melihat-lihat katalog produk. Pelanggan melihat item produk di dalam katalog.
Basic Course	: 1. Usecase dimulai ketika meng-klik Pemesanan Spesifikasi Produk untuk item yang dipilih. 2. Pelanggan memilih format berupa lembar spesifikasi atau brosur. 3. Pelanggan memilih cara pengiriman apakah melalui layanan pos elektronik, kantor pos atau pengiriman dalam satu hari. 4. Daftar alamat dari profile ditampilkan. 5. Pelanggan memilih alamat dari daftar alamat-alamat. 6. Alamat telah dikonfirmasi. 7. Pelanggan mengirimkan permintaan. 8. *Usecase* berakhir pada saat pelanggan memberikan nomor konfirmasi.
Alternative course	:

Use case title	: Memesan Spesifikasi Produk Melalui Surat
1. Jalan alternatif dimulai pada langkah no. 4 2. 4a. Alamat tidak ada pada profile. 4b. Menuju ke form membuat profile baru atau membaharui profile yang ada. 3. *Usecase* dimulai lagi pada langkah no.5	
Postconditions	: Pemesanan telah selesai, terkirim, dan siap untuk menunggu kiriman katalog. Pelanggan kembali pada katalog yang dibuka terakhir kali.
Uses/extends	: Perluasan UC 05.1, Pemesanan Spesifikasi Produk
User implementation requests : Tidak ada	
Frequency	: Terjadi antara 17-100 sesi pelanggan di website.
Unresolved issues	: Tidak ada
Authority	: Mike Danseglio
Modification history	: 6 Desember, 2006
Author	: Heidi Steen
Description	: Versi Inisial

b. Desain Logikal.

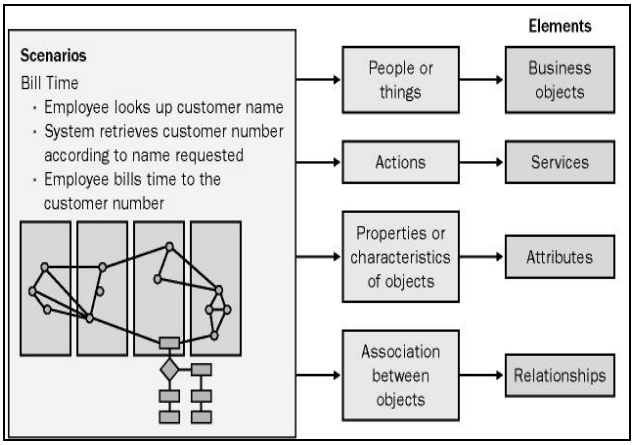
Aktivitas tim pada desain logikal adalah memecah permasalahan yang muncul menjadi unit-unit kecil yang disebut dengan modul. Modul merupakan unit logikal yang digunakan untuk membuat abstraksi usecase dan skenario yang dibuat pada desain konseptual. Untuk setiap modul yang dihasilkan, tim akan mengidentifikasi *objects*, *services* dan *attributes* dan *realationship*. Tim juga mengidentifikasi kandidat teknologi

untuk solusi yang akan dirancang selama desain logikal.

Tim akan melakukan beberapa pertimbangan diantaranya adalah pertimbangan bisnis, arsitektur enterprise, dan teknologi yang dipilih. Pertimbangan bisnis akan sangat berhubungan erat dengan :

- *Feasibility* : menetapkan apakah teknologi sesuai dengan keperluan bisnis dan sesuai dengan *requirement*.
- *Product Cost* : biaya produksi, termasuk didalamnya adalah *developer*, *server*, *reseller license* dan biaya *upgrade.* Tim juga mempertimbang- kan kebutuhan awal *hardware* dan *software*, *support, infrastruktur,* dan *training*.
- *Experience* : tenaga ahli yang dimiliki untuk kebutuhan *training,* konsultasi dan tingkat kenyamanan.
- *Return Of Investment* : investasi harus memiliki keterkaitan dengan pengembalian investasi.
- *Maturity* : produk yang dapat diterima oleh masyarakat, stabil, mudah digunakan dan adanya dukungan sumberdaya.
- *Supportability* : pemilihan teknologi yang sesuai dan mendukung pengembangan solusi jangka panjang.

Tim juga harus mempertimbangkan teknologi yang dipilih, beberapa pertimbangan teknologi adalah masalah yang berhubungan dengan keamanan, standar layanan interaksi, akses data, penyimpanan data, layanan sistem, perangkat pengembangan, dan operating sistem. Selanjutnya melakukan identifikasi terhadap objek bisnis. Objek dapat berupa orang atau berbagai hal yang dideskripsikan didalam *usage scenario*. Objek merupakan dasar dari *services, attributes* dan *relations* .



Gambar 10. Analisa Desain Logikal.

c. Desain Fisikal.

Desain fisikal akan di lakukan setelah semua anggota tim setuju bahwa mereka memiliki informasi yang cukup dari desain logikal yang telah diselesaikan. Tim pengembang akan menerapkan teknologi yang telah dipertimbangkan sebelumnya dengan batasan-batasan pada konseptual dan logikal desain. Tim akan memulai mendiskripsikan komponen, layanan, dan teknologi dari perspektif kebutuhan pengembangan. Fisikal desain mendifinisikan bagian dari solusi yang akan dikembangkan, bagaimana solusi harus dikembangan.

Desain fisikal bukan aktivitas pemrograman (*coding*) atau penyebaran teknologi, melainkan merancang spesifikasi komponen secara detil untuk pengembangan, menetapkan dimana komponen akan ditempatkan dan mengidentifikasi teknologi yang dapat digunakan untuk pengembangan solusi. Aktivitas pada desain fisikal adalah :

- Mengidentifikasi teknologi yang sesuai untuk pengembangan dengan melakukan evaluasi terhadap teknologi yang terbaik.
- Mentransformasikan desain logikal kedalam metode desain fisikal.
- Menyedikan *baseline* untuk proses pengembangan, sebagai tambahan pada fase ini adalah dengan pembuatan model dan strategi,

mendefinisikan aturan dalam proses pengembangan dan tanggung jawab personel pada peran masing-masing.

- Mendefinisikan bilamana rencana proyek disetujui dan *milestone* telah tercapai.

Pada akhir fase desain fisikal, anggota tim proyek memiliki dokumentasi yang cukup lengkap untuk mulai membangun solusi. Beberapa yang dapat dihasilkan pada fase ini adalah :

- Diagram-diagram kelas dari solusi.
- Model komponen, sequence diagram, dan diagram aktivitas.
- Skema database dari solusi.
- *Baseline* model penyebaran yang menyediakan topologi jaringan, lokasi dan interkoneksi *hardware*, topologi data dan komponen, lokasi komponen, layanan dan penyimpanan data .
- Spesifikasi komponen yang terdiri dari struktur dan antarmuka.
- Strategi pemaketan dan distribusi yang mengidentifikasi layanan yang akan dipaketkan bersama dan bagaimana komponen didistribusi melalui topologi jaringan.
- Model pemrograman dan dokumentasi kode.

Dengan menerapkan tiga tahapan desain pada fase planning maka kebutuhan-kebutuhan perancangan perangkat lunak telah didefinisikan dengan lengkap dan jelas. Tim pengembang dapat memulai fase-fase berikutnya sampai solusi dapat distribusikan pada lokasi yang telah direncanakan.

DAFTAR PUSTAKA

1. Preesman, Roger S. 1994. 'Software Engineering : A Practitioner's Approach'. Third Ed., McGraw-Hill International.
2. Sommerville, Lan., 2001, 'Software Engineering ', 6th Addison Wesley.
3. _____,2003, Self Paced Training Kit, 'Analyzing Requirement and Defining Microsoft.NET Solution Architecture, Microsoft Press.
4. _____, Spiral Model Image, http://www.maxwideman.com/papers/linearity/spiral.htm