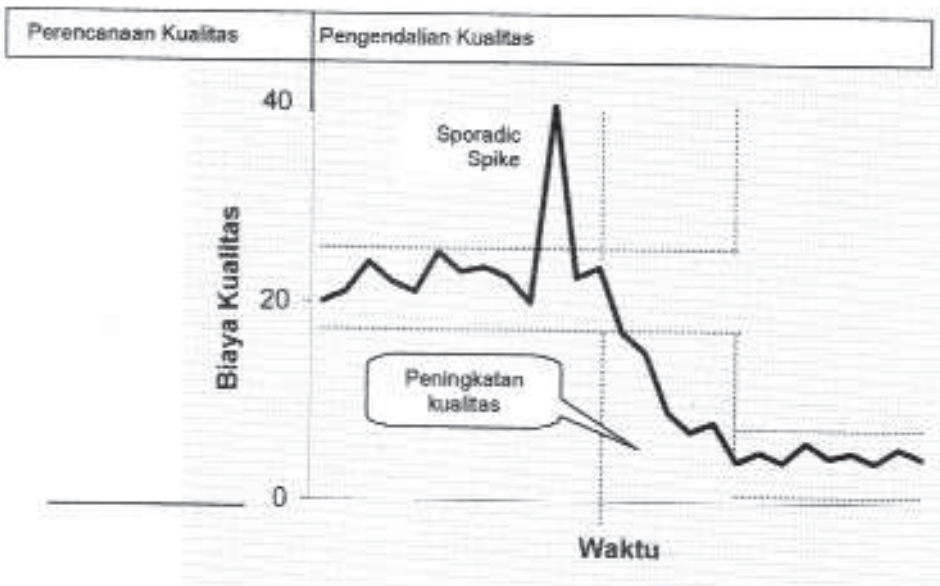
BAB II

LANDASAN TEORI

2.1 Konsep Manajemen Kualitas

Menurut Gaspersz (Gaspersz, 2001, p.5), manajemen kualitas terpadu didefinisikan sebagai suatu cara meningkatkan performansi secara terus-menerus (*continuous performance improvement*) pada setiap level operasi atau proses, dalam setiap area fungsional dari suatu organisasi, dengan menggunakan semua sumber daya manusia dan modal yang tersedia.

Juran sangat terkenal dengan konsep trilogy kualitas, yaitu : perencanaan kualitas (*quality planning*), pengendalian kualitas (*quality planning*), dan perbaikan atau peningkatan kualitas (*quality improvement*). (Gryna, 2001, p.11)



Gambar 2.1 Juran Trilogy Diagram (Gryna, 2001, p.1)

Proses-proses yang ada di dalam konsep Juran tersebut diuraikan dalam Tabel 2.1.

Tabel 2.1 Proses-proses dalam manajemen kualitas menurut Diagram Trilogi Juran (Gryna, 2001, p.11)

PERENCANAAN KUALITAS	PENGENDALIAN KUALITAS	PENINGKATAN KUALITAS
-Membangun proyek	- Memilih subyek kontrol	- Membuktikan kebutuhan
- Identifikasi konsumen	- Membuat pengukuran	- Mengidentifikasi Proyek
- Menemukan kebutuhan konsumen	- Membangun standar penampilan	- Organisasi tim proyek
- Membangun produk	- Mengukur penampilan yang sebenarnya	- Diagnosa kasus
- Membangun proses	- Membandingkan dengan standard	- Menyediakan dan menguji cara memperbaiki
- Mengembangkan kendali proses, transfer ke operasi.	- Mengambil tindakan terhadap perbedaan	- Berhubungan dengan kesulitan untuk berubah dan mengontrol untuk mencapai hasil.

2.2 Konsep Six Sigma

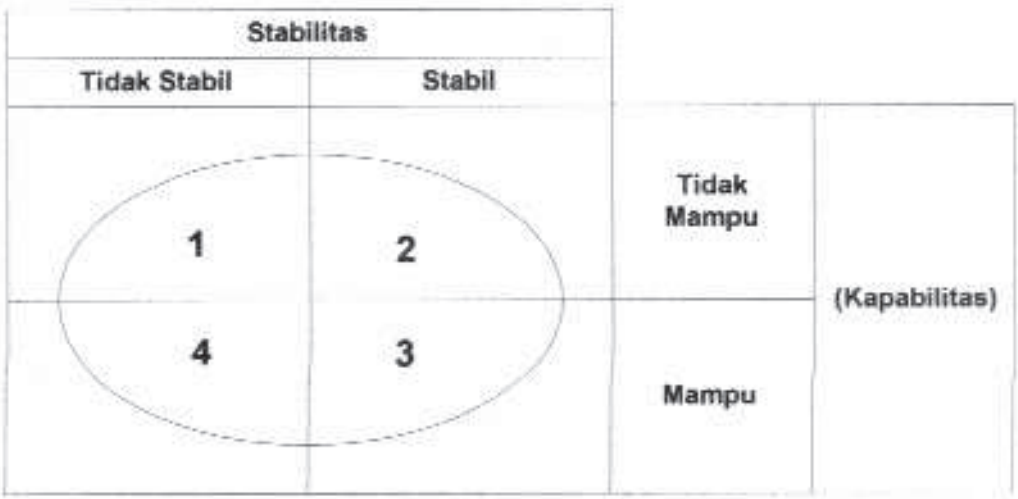
2.2.1 Definisi Six Sigma

Menurut Pearce dan Robinson (Pearce, 2003, p.330), Six-Sigma merupakan suatu pendekatan dengan ketepatan dan kemampuan analisis yang tinggi terhadap kualitas dan peningkatan yang terus-menerus dengan tujuan untuk meningkatkan keuntungan melalui pengurangan kerusakan (*defect*), peningkatan hasil, peningkatan kepuasan konsumen, dan kinerja terbaik (*best-in-class*).

Key success factors yang membedakan Six-Sigma dengan TQM adalah sebagai berikut :

1. Memperdalam pengertian mengenai konsumen dan produk atau jasa yang disediakan.
2. Menekankan pada ilmu statistik dan pengukuran.
3. Memiliki pelatihan yang sangat teliti dan terstruktur.
4. Metodologi yang ketat dan berfokus pada proyek.
5. Memperkuat doktrin yang dianjurkan oleh Juran seperti dukungan manajemen puncak dan pembelajaran yang terus-menerus.

Menurut Dr. William Edwards Deming (Gaspersz, 2002, p.201), pada dasarnya pengendalian dan peningkatan proses industri mengikuti konsep siklus hidup proses (*process life cycle*) seperti ditunjukkan pada Gambar 2.2.



Gambar 2.2. Siklus Hidup Proses (Gaspersz, 2002, p. 202)

Interpretasi dari siklus hidup proses dapat dilihat pada Tabel 2.2.

Tabel 2.2. Analisis Sistem Sepanjang Siklus Hidup Proses. (Gaspersz, 2002, p.203)

Status Proses				
No.	Stabilitas	Kapabilitas	Situasi	Analisis
1.	Tidak	Tidak	▪ Keadaan proses di luar pengendalian. ▪ Proses akan menghasilkan produk cacat terus-menerus (keadaan kronis).	Sistem berada dalam kondisi paling buruk.
2.	Ya	Tidak	▪ Keadaan proses berada di dalam pengendalian. ▪ Proses masih menghasilkan produk cacat.	Sistem berada dalam status menuju peningkatan kualitas global.
3.	Ya	Ya	▪ Keadaan proses berada di dalam pengendalian. ▪ Proses tidak menghasilkan produk cacat (*zero defects*).	Sistem berada dalam kondisi paling baik, merupakan target dari program Six Sigma.
4.	Tidak	Ya	▪ Proses berada di luar pengendalian. ▪ Proses menimbulkan masalah kualitas secara sporadis.	Sistem tidak dapat diperkirakan (*unpredictable*) dan tidak diinginkan (*undesirable*) oleh pihak manajemen.

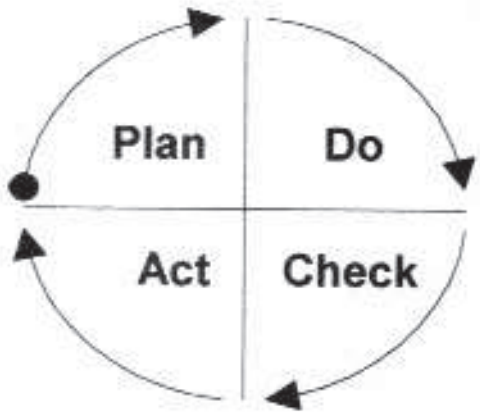
Dari Gambar 2.2 dan Tabel 2.2., kita mengetahui bahwa target dari program Six Sigma adalah membawa proses industri untuk beroperasi pada kondisi no. 3, yaitu proses industri yang memiliki stabilitas dan kapabilitas, sehingga mencapai tingkat kegagalan nol (*zero defect oriented*).

2.2.2 Model Pengembangan Six Sigma

Ada banyak model pengembangan yang diterapkan pada proses-proses selama bertahun-tahun sejak dimulainya pergerakan kualitas. Pada umumnya, model-model ini berdasarkan pada langkah-langkah yang diperkenalkan oleh Deming, yaitu *Plan-Do-Check-Act* atau P-D-C-A, yang menjelaskan logika dasar dari proses pengembangan *database*.

- ***Plan***. Meninjau performansi saat ini dari segi masalah dan kesenjangan. Mengumpulkan data mengenai masalah-masalah utama. Mengidentifikasi dan menargetkan akar penyebab dari masalah. Memikirkan solusi yang mungkin, dan merencanakan pengujian implementasi dari solusi dengan potensial tertinggi.
- ***Do***. Menuntun solusi yang telah direncanakan.
- ***Check***. Mengukur hasil test untuk melihat apakah hasil yang diinginkan telah dicapai. Jika masalah meningkat, cari hambatan yang menghalangi usaha-usaha pengembangan.

- ***Act***. Berdasarkan pengujian solusi dan evaluasi, kembangkan solusi dan untuk dijadikan permanen, dan gabungkan dengan pendekatan yang baru jika dapat diterapkan.



Gambar 2.3 P-D-C-A Cycle (Woodall, 1997, p.164)

Woodall (1997, p.164) menerapkan prinsip P-D-C-A ini dalam proyek IT, seperti digambarkan pada Gambar 2.3.

Langkah pertama, ***Plan***, menanyakan beberapa pertanyaan penting seperti perubahan apa yang diperlukan, hasil apa yang diinginkan, rintangan-rintangan apa yang perlu diatasi, apakah datanya tersedia, dan informasi baru apa saja yang diperlukan.

Do merupakan implementasi perubahan skala kecil atau *pilot test* untuk menyediakan data untuk jawaban.

Check merupakan penilaian dari akibat-akibat perubahan atau test.

Act (*final test*) pertama-tama menanyakan apakah data-data yang ada memperkuat rencana yang dimaksudkan, apakah variabel-variabel lain

mempengaruhi rencana, dan apakah resiko dalam cara bekerja dianggap penting dan berharga. Kemudian, berdasarkan pada jawaban-jawaban ini, proyek atau tugas dimodifikasi dan dipindahkan ke tahap plan sekali lagi, di mana terjadi perulangan, peningkatan pengetahuan, dan pengembangan yang lebih lanjut. Secara ideal, proses ini akan berlangsung untuk jangka waktu yang tidak terbatas.

Menurut Pande, Neuman, dan Cavanagh (Pande, 2000, p.69), ada 5 fase siklus pengembangan banyak dipakai dalam organisasi yang menerapkan Six Sigma.: *Define, Measure, Analyze, Improve, and Control*, atau DMAIC. Seperti model pengembangan lain, DMAIC berdasarkan pada siklus PDCA. Tabel 2.3 menjelaskan aktivitas DMAIC yang utama, dengan membandingkan "Pengembangan Proses" dengan "Rekayasa/Rekayasa Ulang Proses".

Ada beberapa keuntungan potensial dari DMAIC :

1. Melakukan langkah awal yang baru. DMAIC akan membantu menempatkan Six Sigma sebagai model yang berbeda namun menawarkan pendekatan yang lebih baik untuk pengembangan bisnis.
2. Memberikan konteks yang baru terhadap *familiar tools*, sehingga memberi kesempatan bagi karyawan untuk mempelajari dan mempraktekkan alat-alat *familiar* dan menambah alat yang baru.
3. Menciptakan pendekatan yang konsisten.
4. Menempatkan prioritas terhadap Konsumen dan Pengukuran.
5. Menawarkan baik "Pengembangan Proses" maupun "Rekayasa/Rekayasa Ulang Proses" untuk pengembangan.

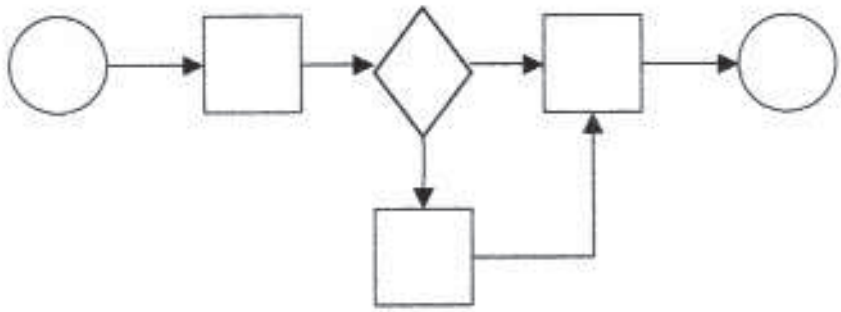
Tabel 2.3. Proses Pengembangan Six Sigma (Pande, 2000, p.71)

Proses Pengembangan Six Sigma		
	Peningkatan Proses	**Rekayasa/Rekayasa Ulang Proses**
Define	✓ Identifikasi Masalah ✓ Definisi kebutuhan ✓ Menentukan tujuan	✓ Identifikasi masalah spesifik atau masalah besar ✓ Definisi tujuan/mengubah visi ✓ Klarifikasi ruang lingkup & kebutuhan konsumen
Measure	✓ Validasi Masalah/Proses ✓ Melengkapi tujuan/definisi masalah ✓ Mengukur langkah-langkah kunci/input	✓ Mengukur performansi untuk kebutuhan ✓ Mengumpulkan data yang menyebabkan efisiensi proses
Analyze	✓ Membangun hipotesis penyebab masalah ✓ Identifikasi akar penyebab masalah yang vital ✓ Validasi hipotesis	✓ Identifikasi "praktek terbaik" ✓ Menilai rancangan proses - Menambah nilai/tidak - Kemacetan/terputus - Jalur pengganti ✓ Melengkapi persyaratan
Improve	✓ Membangun ide untuk menghilangkan akar masalah ✓ Menguji solusi ✓ Standardisasi solusi/hasil pengukuran	✓ Merancang proses baru - Membangun asumsi - Menerapkan kreativitas - Prinsip-prinsip workflow ✓ Implementasi proses, struktur, dan sistem baru
Control	✓ Membangun standar ukuran untuk mempertahankan kualitas ✓ Memperbaiki masalah jika diperlukan	✓ Membangun ukuran dan review untuk mempertahankan kualitas ✓ Memperbaiki masalah jika diperlukan

2.2.3 Analytical Tools untuk Six Sigma

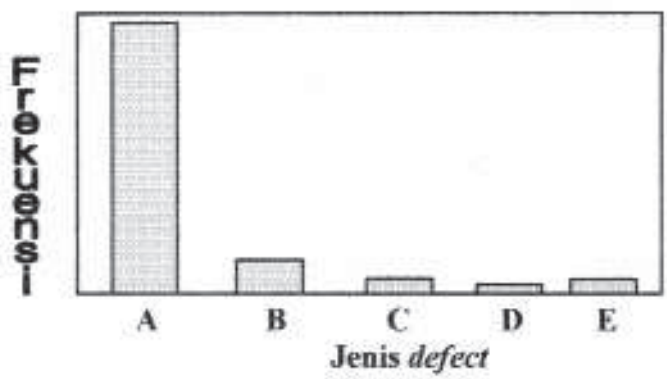
Menurut Chase, Aquilano, dan Jacobs (Chase, 2001, p.272), terdapat beberapa *analytical tools* untuk Six Sigma :

1. *Process Flowchart*, merupakan gambar yang menjelaskan langkah-langkah utama, cabang-cabang, dan hasil terakhir dari suatu proses.



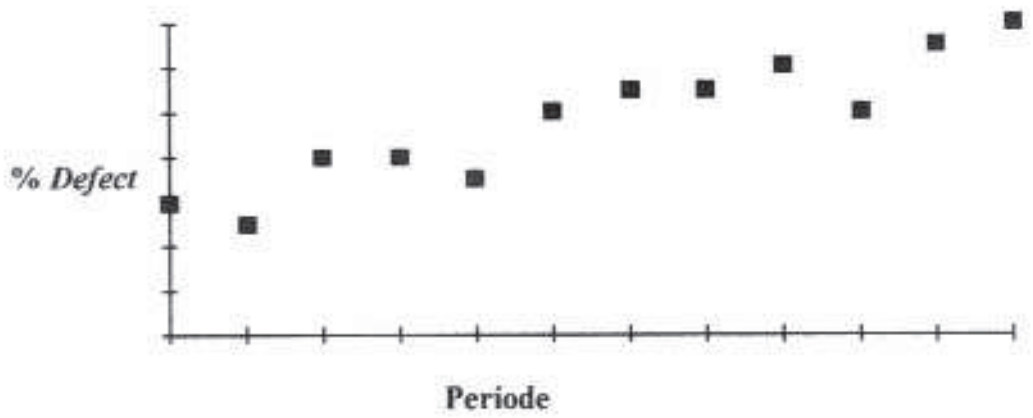
Gambar 2.4 *Process Flowchart* (Chase, 2001, p.272)

2. *Pareto Analysis*, merupakan pendekatan yang terkoordinasi untuk mengidentifikasi, penyusunan, dan pekerjaan untuk secara permanen menghilangkan *defect*. Berfokus pada sumber-sumber kegagalan yang penting. 80/20 *rule* : 80% masalah disebabkan oleh 20% kasus.



Gambar 2.5 Pareto Analysis (Chase, 2001, p.272)

3. *Run Chart*, merupakan grafik yang bersifat sekuensial terhadap waktu yang menunjukkan gambaran nilai dari suatu karakteristik.

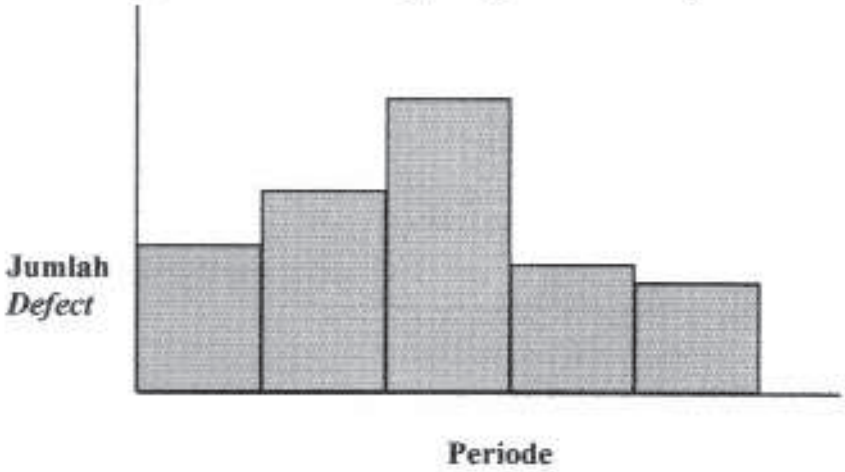


Gambar 2.6 Run Chart (Chase, 2001, p.272)

4. *Data Collection*, selalu memiliki persetujuan dan alasan yang jelas mengenai data yang dikumpulkan. Mempersiapkan dari awalnya strategi untuk mengumpulkan dan menganalisa data. Pertanyaan yang mungkin diajukan untuk pengumpulan data :

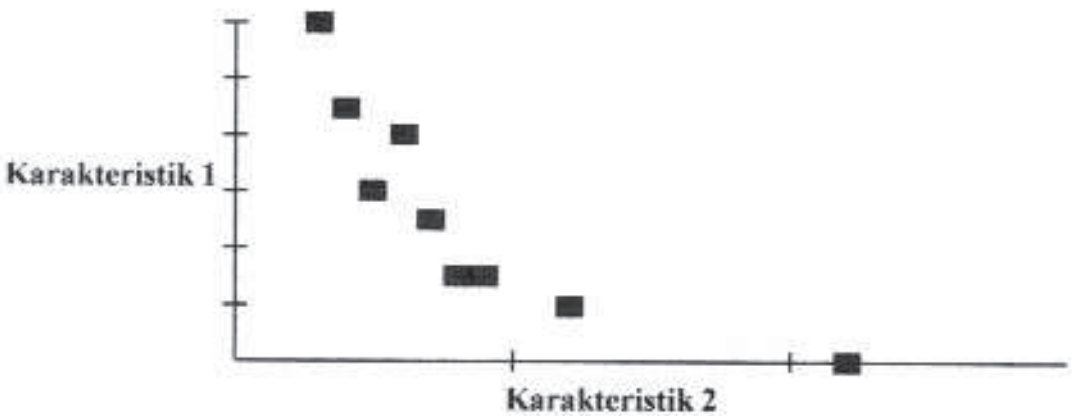
- Mengapa ?
- Apa?
- Di mana?
- Berapa banyak?
- Kapan?
- Bagaimana?
- Berapa lama?

5. *Histogram*, merupakan suatu distribusi yang menunjukkan frekuensi kejadian antara data *high range* dan *low range*.



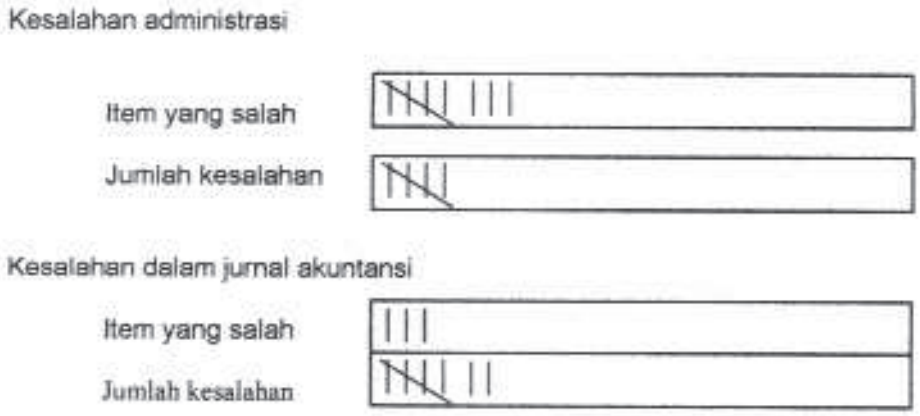
Gambar 2.7 Histogram (Chase, 2001, p.272)

6. *Scatter Diagram*, dikenal juga sebagai diagram korelasi, merupakan grafik yang menggambarkan nilai dari suatu karakteristik dibandingkan dengan karakteristik lain.



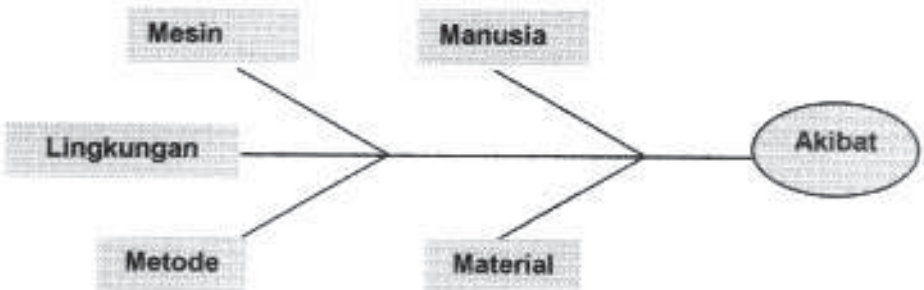
Gambar 2.8 *Scatter Diagram* (Chase, 2001, p.272)

7. *Check Sheet*, merupakan metode yang terorganisir untuk merekam data, dapat digunakan untuk mengingat *defect* atau untuk meyakinkan bahwa data telah dikumpulkan dengan cara yang benar.



Gambar 2.9 *Check Sheet* (Chase, 2001, p.272)

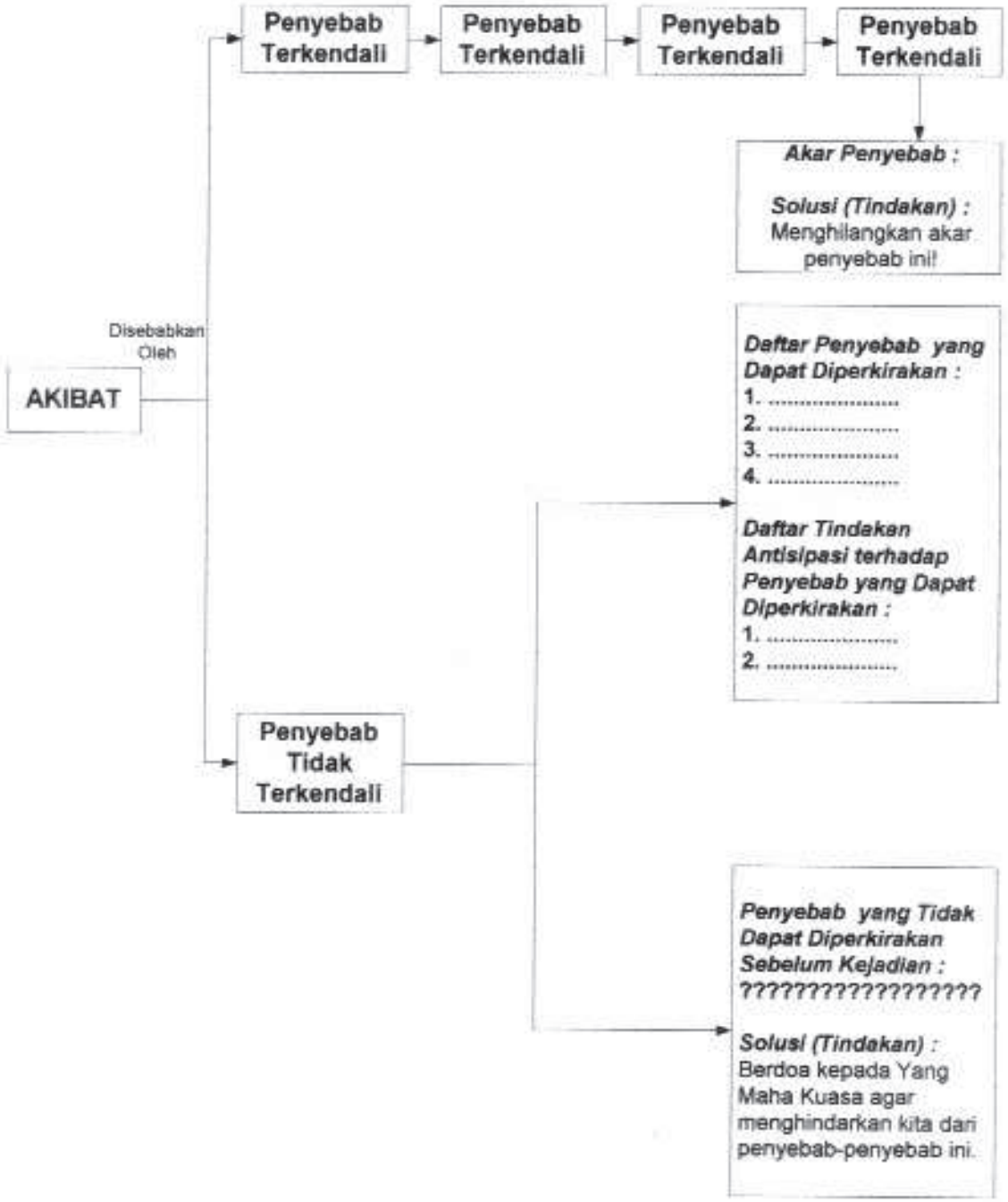
8. *Cause and Effect* atau *Fishbone Diagram*, merupakan alat yang menggunakan deskripsi grafis dari elemen proses untuk menganalisa sumber-sumber potensial dari variasi proses.



Gambar 2.10 Fishbone Diagram (Chase, 2001, p.272)

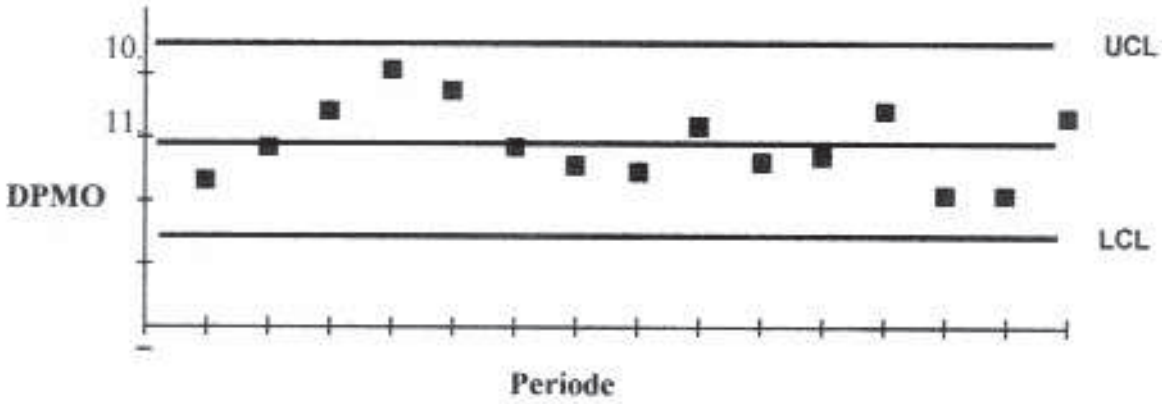
Suatu solusi masalah yang efektif adalah apabila kita berhasil menemukan sumber-sumber dan akar-akar penyebab dari masalah itu, kemudian mengambil tindakan untuk menghilangkan akar-akar penyebab masalah itu. Untuk dapat menemukan akar penyebab dari suatu masalah, maka kita perlu memahami dua prinsip yang berkaitan dengan hukum sebab-akibat (Gaspersz, 2002, p.237). Prinsip-prinsip tersebut adalah :

- Suatu akibat terjadi atau ada hanya jika penyebab dari kejadian tersebut itu berada pada titik yang sama dalam suatu ruang dan waktu.
- Setiap akibat mempunyai paling sedikit dua penyebab dalam bentuk: penyebab yang dapat dikendalikan (*controllable causes*) dan penyebab yang tidak dapat dikendalikan (*uncontrollable causes*). Penyebab yang dapat dikendalikan berarti penyebab itu berada dalam lingkut tanggung jawab dan wewenang kita sehingga dapat mengambil tindakan untuk menghilangkan penyebab itu. Sebaliknya penyebab yang tidak dapat dikendalikan berada di luar pengendalian kita. Penyebab yang tidak dapat dikendalikan terdiri dari paling sedikit dua penyebab, yaitu: penyebab yang dapat diperkirakan (*predictable causes*) sehingga kita dapat mengantisipasinya, dan penyebab yang tidak dapat diperkirakan karena belum ada referensi atau pengetahuan tentang kejadian itu sebelumnya. Untuk lebih jelasnya dapat dilihat pada Gambar 2.11



Gambar 2.11. Diagram untuk Mengidentifikasi Penyebab-Penyebab dari Suatu Masalah Kualitas (Gaspersz, 2002, p.239)

9. *Control Charts*, merupakan grafik yang bersifat sekuensial terhadap waktu, yang menunjukkan nilai-nilai dari statistik, termasuk garis pusat dan satu atau lebih batasan kendali yang diperoleh secara statistik.



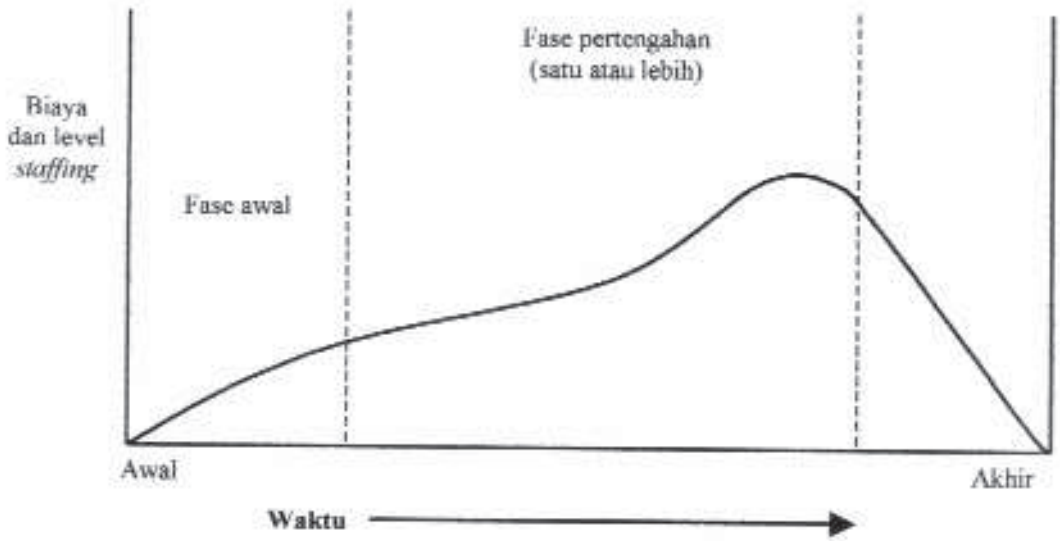
Gambar 2.12 Control Charts (Chase, 2001, p.272)

2.3 Konsep Manajemen Proyek

Menurut *Project Management Institute* (PMI, 2000, p.6), manajemen proyek adalah aplikasi pengetahuan, keterampilan, peralatan dan teknik-teknik untuk memproyeksikan aktivitas untuk memenuhi persyaratan dan kebutuhan proyek. Manajemen proyek dikerjakan melalui proses-proses seperti: *initiating, planning, executing, controlling, dan closing.*

Konsep manajemen proyek merupakan alat yang tepat untuk menentukan teknik-teknik dan aktivitas yang dibutuhkan unruk menyelesaikan suatu proyek. Sedangkan seberapa baik kualitas dari suatu proyek yang sedang berjalan saat ini belum dapat ditentukan dengan tepat oleh konsep ini. Analisa mengenai tingkat kualitas proyek dan penyebab masalah kualitas lebih tepat dilakukan dengan metode Six Sigma dengan target untuk mencapai tingkat kegagalan nol (*zero defect*).

Definisi siklus hidup proyek (*project life cycle*) adalah pekerjaan teknis apa saja yang harus dilakukan dan siapa yang harus dilibatkan pada masing-masing fase. Pada tahap awal, probabilitas untuk berhasil dalam menyelesaikan proyek adalah yang terendah, sementara resiko dan ketidakpastian adalah yang tertinggi. Probabilitas ini biasanya bertambah secara progresif ketika proyek dilanjutkan ke tahap berikutnya. Dalam *project life-cycle* seperti ditunjukkan pada Gambar 2.13 (PMI, 2000, p.13), biaya proyek dan *staffing level* rendah pada permulaan proyek, dan bertambah besar seiring bertambahnya waktu, dan berkurang drastis ketika proyek mendekati tahap penyelesaian.



Gambar 2.13 Project Life Cycle (PMI, 2000, p.14)

Menurut Schwalbe (2000, p.184-185), seringkali sulit untuk mengerti sepenuhnya mengenai dimensi *performance* dari proyek IT. Bahkan jika teknologi *hardware, software*, dan jaringan tidak berubah untuk sementara waktu, seringkali

sulit bagi *user* untuk menjelaskan secara tepat mengenai apa yang mereka inginkan dalam proyek IT. Aspek penting dari proyek IT yang mempengaruhi kualitas meliputi *functionality and features, system outputs, performance,* dan *reliability and maintainability.*

1. ***Functionality* dan *Features.***

   Functionality merupakan tingkat dimana suatu sistem menampilkan fungsi yang diinginkan. *Features* merupakan karakteristik khusus yang menarik bagi *user*. Sangat penting untuk mengklarifikasi apa *function* dan *features* yang perlu ditampilkan oleh sistem, dan *function* dan *features* apa yang bersifat *optional*.

2. ***System outputs.***

   System outputs merupakan *screen* dan laporan yang dihasilkan oleh sistem. Sangatlah penting untuk mendefinisikan dengan jelas mengenai bagaimana tampilan *screen* dan laporan untuk suatu sistem. Dapatkah *user* menginterpretasikan *output* tersebut? Apakah *user* mendapatkan laporan yang mereka perlukan dalam format yang sesuai?

3. ***Performance.***

   Performance berarti seberapa baik suatu program menampilkan kebutuhan *user*. Untuk merancang sistem dengan penampilan berkualitas tinggi, ada banyak hal yang perlu diperhatikan, seperti: Berapa banyak data dan transaksi yang harus ditangani oleh sistem? Berapa banyak *user* yang menggunakan sistem? Seberapa cepat waktu respon untuk aspek-aspek yang berbeda dari

sistem dalam lingkungan yang berbeda? Masalah lain dalam hal *performance* yang mungkin sulit ditangani adalah fakta bahwa beberapa laporan menghasilkan data yang tidak konsisten.

4. ***Reliability* dan *maintainability*.**

   Reliability merupakan kemampuan program untuk menampilkan sesuai dengan harapan dalam kondisi normal tanpa kegagalan yang tidak dapat ditolerir. *Maintainability* merupakan kemudahan dalam memelihara program. Sebagian besar proyek IT tidak dapat menghasilkan 100 persen *reliability*, tetapi *user* perlu mendefinisikan apa yang menjadi kebutuhan dan keinginan mereka.

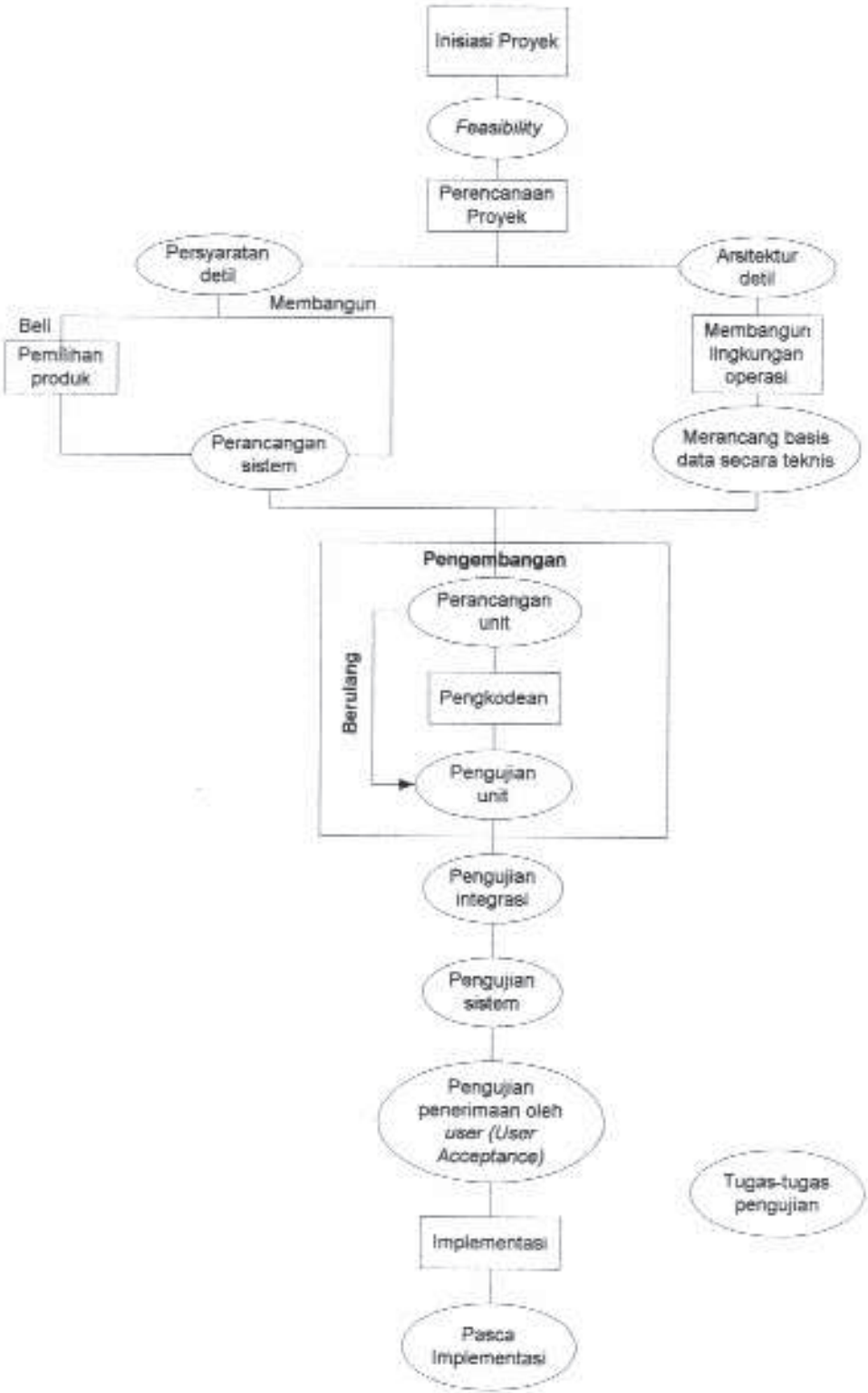
2.3 Konsep Teknologi Informasi

Menurut Woodall (Woodall, 1997, p.44), ada beberapa profesi IT yang berhubungan dengan pengembangan program aplikasi, yaitu :

- Manajer tim Proyek. Bertanggung jawab terhadap tim IT yang terlibat dalam proyek. Berhubungan dengan konsumen atau pengguna program aplikasi. Menyiapkan laporan status dan laporan kemajuan.
- *System analyst*, yaitu seseorang yang memiliki kemampuan analitis, menemukan pemecahan masalah, berhubungan dengan *end user*, menghasilkan spesifikasi kebutuhan, dan membantu dalam pengujian dan rencana pengembangan system.

- *Programmer*, bertugas merancang, memprogram, dan menguji program aplikasi.
- *Data analyst*, bertugas merencanakan dan merancang basis data secara teknis. Memelihara dan menjaga keamanan data atau program aplikasi yang berhubungan dengan data.

Menurut Schwalbe (Schwalbe, 2002, p.193), banyak professional IT yang menganggap pengujian dilakukan pada tahap yang mendekati tahap akhir pengembangan proyek IT. Sebenarnya, pengujian perlu dilakukan pada hampir setiap tahap dari siklus hidup pengembangan proyek IT. Gambar 2.14 menunjukkan siklus hidup pengembangan piranti lunak.



Gambar 2.14. Tugas-tugas pengujian dalam Siklus Hidup Pengembangan Piranti Lunak. (Schwalbe, 2002, p.194)