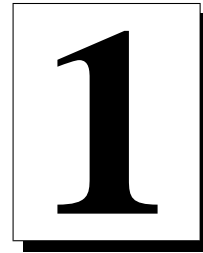


Dokumentasi dalam pengembangan Sistem Informasi

I Made Wiryana

11 Oktober 2004



Kebutuhan dokumentasi

Suatu proses bisnis (dalam hal ini bisnis bukan hanya perdagangan saja, melainkan suatu proses pada pelaksanaan manajemen) penting untuk implementasi suatu program atau sistem. Bahaya yang dihadapi oleh proyek Intranet dan Internet adalah kecepatan yang harus digunakan pada saat implementasi. Pemendekan siklus pengembangan untuk suatu proyek bukanlah merupakan suatu alasan yang baik.

Pada suatu bagian sistem informasi, menaikkan kualitas proses biasanya melibatkan elemen berikut ini :

- **Metodologi .**

Suatu cara, metoda, untuk mencapai tujuan. Suatu metodologi berlaku secara umum, dengan perencanaan tingkat tinggi, dan digunakan sebagai landasan setiap proyek. Ada beberapa metoda khusus untuk beberapa jenis proyek yang khusus, seperti metodologi untuk Internet atau Intranet.

- **Dokumentasi .**

Dokumen khusus, yang pada awal proyek akan menerangkan secara garis besar. Yang akan dilengkapi pada setiap proyek yang dilaksanakan. Contoh dokumentasi adalah : Functional Specification, Cost-benefit Analysis, and Return of Investment.

- **Standard .**

Panduan yang disusun dan digunakan pada suatu institusi untuk menyelesaikan suatu pekerjaan. Contoh standard ini adalah : kesepakatan penamaan untuk berbagai macam kode, kesepakatan layar GUI, kesepakatan data modelling. Standard ini penting karena merupakan landasan pengembangan sebagai kerangka kerja, komunikasi. Juga untuk mengontrol kualitas serta menjaga kontinuitas pengembangan.

Hal penting dari suatu Business Process Reengineering (BPR) adalah mengoptimasi metoda untuk melaksanakan suatu pekerjaan, yang dapat dipakai ulang

untuk proyek mendatang. Setelah suatu metodologi ditentukan, akan lebih mudah untuk para manager memahami pola antar proyek dan menentukan mana yang bekerja dan mana yang tidak. Akan lebih mudah mengetahui mengapa suatu proyek menjadi gagal, dan mengetahui titik penting yang mengakibatkan kesuksesan suatu proyek. Kesuksesan inilah yang sering disebut dengan *best practices*. Bila hal ini telah didefinisikan maka dapat digunakan berulang sehingga dikenal dengan istilah *reusable processs* .

Untuk kesuksesan suatu proyek, disain dan implementasi harus dilakukan dengan sama baiknya. Pada suatu BPR, problem sering kali didefinisikan oleh manager tingkat tinggi, yang tak mengetahui bagaimana pekerja sebenarnya mengerjakan hal tersebut. Pada kasus lain, seringkali konsultan melakukan proses BPR tanpa memahami dengan benar bagaimana proses tersebut dilaksanakan.

Solusi dari permasalahan ini adalah kesepakatan penggunaan BPR dan keterlibatan setiap personal. Proses harus dievaluasi oleh developer, analis, arsitek sistem, manager, dan juga oleh pengguna. Pendekatan ini lebih kepada pendekatan *bottom up* daripada *top down* . BPR membutuhkan sejumlah konsensus yang harus disepakati untuk dilaksanakan secara bersama-sama.

1.1 Dokumentasi proyek

Dokumentasi adalah suatu hal yang pertama-tama harus ditentukan dan diselesaikan. Hal yang penting agar dokumentasi dapat disusun dengan sukses adalah, dilakukan dengan cara mengintegrasikan dokumentasi ini dengan metodologi, sehingga proses dokumentasi dilakukan ketika setiap langkah development dilakukan, daripada melakukannya setelah selesai. Bentuk dasar dari dokumentasi ini sebaiknya juga dilakukan untuk proyek-proyek yang lainnya.

Pada suatu proyek biasanya terdapat 6 proses yang saling terkait dan dinamis. Proses ini adalah :

- Pendefinisian
- Perencanaan
- Organisasi
- Pengawasan
- Penyelesaian
- Leading

Setiap proses akan memiliki keluaran yang akan menjadi masukan bagi proses yang lainnya. Proses-proses ini memberikan beberapa keuntungan termasuk :

- Mengetahui dampak teknologi dan bisnis
- Menghitung estimasi biaya sesungguhnya

- Menentukan tingkat usaha
- Mencapai suatu penyelesaian yang paling efektif biayanya.
- Memilih perangkat bantu dan teknik terbaik

1.1.1 Pendefinisian

Dengan mendefinisikan proyek dengan tetap, diharapkan proyek dapat mulai dan diakhiri dengan biaya yang paling efektif. Termasuk menjawab : **who, what, when, where, why and how** dari pelaksanaan proyek tersebut. Perangkat bantu untuk melaksanakan tugas ini disebut dengan **Statement of the Works (SOW)** .

SOW adalah kesepakatan antar client dan developer. Dokumen ini ditulis berdasarkan perspektif bisnis dan teknis yang berisi topik-topik berikut ini :

- Pengantar (misal informasi latar belakang)
- Tujuan dan obyektif (misal cost, jadwal, dan kualitas)
- Scope (misal, aplikasi HTML atau VRML)
- Assumsi (misal kemampuan penanganan peningkatan traffic jaringan)
- User
- Sumber daya (misal spesialis jaringan, programmer)
- Milestone untuk penjadwalan (misal waktu akhir testing)
- Pembiayaan (biaya langsung dan overhead)
- Amandement (definisi ulang dari penyerahan pekerjaan)
- Tanda tangan (manajemen senior dan komunitas pengguna)

SOW memberikan keuntungan ketika digunakan untuk memulai suatu proyek Intranet. Yaitu :

- Menjelaskan biaya dan jadwal juga asumsi utama tentang proyek
- Menjelaskan peranan dan tanggung jawab.
- Mengukuhkan definisi hal yang akan dicapai proyek Intranet tersebut.
- Mendorong diselesaikannya proyek tersebut, karena adanya kesepakatan tertulis dalam dokumen tersebut (tanda tangan).

Di samping itu SOW ini akan membantu menentukan tanggung jawab sekuriti pada tingkat tinggi, perawatan dokumentasi, perangkat lunak, data, perangkat keras, dan pengelolaan sistem. Dengan kata lain akan mendefinisikan siapa yang berperan sebagai web-masters, document-master, dan document-owners. SOW juga mencegah permasalahan yang timbul di tahapan berikutnya dari pengembangan sistem.

1.1.2 Perencanaan

SOW menjabarkan biaya secara kasar, penjadwalan, kualitas, dan sumberdaya manusia pada suatu proyek. Dengan informasi ini perencanaan dilakukan dengan berdasarkan pada informasi ini. Perencanaan sebagai langkah berikutnya meliputi 6 tahapan yang dapat dilaksanakan secara berurutan ataupun paralel :

- Menyusun **Work Breakdown Structure (WBS)**
- Mengestimasi waktu pelaksanaan proyek
- Mengalokasikan sumber daya
- Menghitung pembiayaan
- Menyusun jadwal kerja
- Pengelolaan resiko

Menyusun WBS

Pada dasarnya WBS merupakan suatu daftar yang bersifat top down dan secara hirarkis menerangkan komponen-komponen yang harus dibangun, dan pekerjaan yang berkaitan dengannya. Sebagai contoh pada tabel di bawah ini

Nomor Pekerjaan	Pekerjaan
1.0	Home Page
1.1	Penentuan isi
1.2	Penentuan format dan layout
1.3	Penyusunan homepage
2.4	Sekuriti
2.5	Otentikasi
2.6	Pembatasan akses
2.7	Firewall
2.8	Event Logging
2.9	Enkripsi
2.10	Kebijaksanaan
3.11	Monitoring
3.12	Reporting (pelaporan)
3.1.1	Baseline dan trend analysis
3.2	Metrics
3.2.1	Penentuan metoda untuk menghitung waktu koneksi
3.2.2	Penentuakn metoda untuk menghitung throughput rate
3.2.3	Penentuan metoda untuk menghitung waktu respon
4.4	Server
4.5	Perangkat keras Server
4.1.6	Penentuan kebutuhan perangkat keras server
4.1.7	Pemilihan perangkat keras server
4.1.8	Instalasi perangkat keras server
4.9	Perangkat lunak
4.2.10	Directory listing - struktur
4.2.11	Platform
4.2.12	IP/adres, URL dan nama domain
4.13	Client
4.3.1	Perangkat keras
4.3.1.1	Penentuan kebutuhan perangkat keras client
4.3.1.2	Pemilihan perangkat keras client
4.3.1.3	Instalasi perangkat keras client
4.3.2	Perangkat lunak
4.3.2.3	Instalasi perangkat lunak
4.3.2.1.4	Instalasi FTP
4.3.2.1.5	Instalasi email
4.3.2.1.6	Instalasi telnet
4.3.2.1.7	Instalasi browser
4.3.2.1.8	Instalasi sistem operasi
4.3.2.9	Konfigurasi perangkat lunak

Model WBS memberikan beberapa keuntungan

- Memberikan daftar pekerjaan yang harus diselesaikan

- Memberikan dasar untuk mengestimasi, mengalokasikan sumber daya, menyusun jadwal, dan menghitung biaya
- Mendorong untuk mempertimbangkan secara lebih serius sebelum membangun suatu proyek Intranet.

Mengestimasi waktu pelaksanaan proyek

Dengan memanfaatkan daftar pekerjaan pada WBS, dapat dilakukan pekerjaan memperkirakan waktu yang dibutuhkan untuk menyelesaikan setiap pekerjaan tersebut. Perkiraan dilakukan dengan beberapa pertimbangan : ketersediaan sumber daya dan kompleksitas. Kemudian dijabarkan dalam kalender atau *flow time* . Biasanya optimasi dilakukan secara

- **most optimistic** . Waktu ideal untuk menyelesaikan pekerjaan, diasumsikan segala sesuatunya berjalan lancar, dan sempurna.
- **most likely** . Waktu yang dibutuhkan pada kondisi kebanyakan, tipikal dan normal.
- **most pessimistic** . Waktu yang dibutuhkan ketika keadaan paling sulit terjadi.

Estimasi waktu dilakukan dan dibagi dalam unit (misal 8 jam hari). Estimasi waktu untuk suatu proyek Intranet lebih sulit dari proyek pengembangan aplikasi lainnya. Hal ini karena masih sedikit proyek yang dapat digunakan sebagai patokan menghitung waktu pelaksanaan. Dalam mengestimasi waktu ini juga harus dipertimbangkan beberapa hal, misal pengalaman teknologi server yang digunakan, keahlian Perl, CGI, Java dan HTML, browser, dan juga bekerja dalam lingkungan TCP/IP.

Penentuan resiko

Prioritas penting ditentukan pada setiap proyek, termasuk juga pada proyek Intranet. Sebab seperti halnya Internet ada beberapa permasalahan sekuriti (seperti akses tanpa hak), dan karena adanya banyak komponen pembentuk sistem (misal browser dan server) yang terlibat, resiko dapat menjadi tinggi. Penentuan resiko akan membantu melakukan identifikasi resiko yang dihadapi setiap komponen. Dengan informasi ini seorang manajer proyek dapat menentukan tingkat kepentingan setiap tugas dan menentukan estimasi waktu untuk itu. Manajer proyek dapat berkonsentrasi pada waktu dan sumber daya pada elemen yang terkritis dari penjadwalan.

Menyusun jadwal kerja

Pada dasarnya ada dua jenis model deskripsi penjadwalan :

- **Bar Chart**, yang hanya menerangkan flow time dari setiap pekerjaan dan tanpa keterkaitan antar pekerjaan. Deskripsi ini paling baik digunakan pada presentasi

- **Network diagram**, yang menunjukkan keterkaitan antar tugas dan mengidentifikasi saat kritis pada jadwal.

Suatu network diagram, merupakan cara terbaik untuk merencanakan secara detail, dan mengikuti perkembangan proyek. Diagram ini akan menghubungkan pekerjaan terkait, dan waktu mulai dan berakhirnya dari pekerjaan tersebut. Mengidentifikasi keterkaitan pekerjaan pada proyek Intranet adalah sangat penting sebab komponen-komponen tersebut saling terkait agar dapat bekerja sesuai dengan fungsinya

Mengalokasikan sumber daya

Pada dasarnya harus dilakukan pengimbangan waktu setiap pekerjaan dan ketersediaan dan kemampuan sumber daya. Harus ditentukan *level load* dari sumber daya, agar tak ada personal yang bekerja terlalu berat, dan ada yang terlalu ringan. Pada proyek Intranet hal ini sulit, karena tidak tersedianya sumberdaya manusia yang memiliki keahlian tersebut, oleh sebab itu harus disusun jadwal yang realistis. Dan bahkan mungkin dilakukan revisi penjadwalan.

Menghitung pembiayaan

Yang menjadi permasalahan, apakah biaya yang akan dikeluarkan sesuai dengan SOW. Jika sesuai, maka pekerjaan perencanaan selesai, bila tidak harus dilakukan revisi. Bila memang sulit harus dilakukan negosiasi dengan pihak pemberi kerja. Ketika melakukan perhitungan biaya perlu dipertimbangkan beberapa biaya tersembunyi, misal training, dokumentasi.

1.1.3 Organisasi

Proses ini adalah proses yang melibatkan penyusunan suatu infrastruktur yang akan memaksimalkan efisiensi dan efektifitas ketika melaksanakan proyek. Yang harus dipertimbangkan adalah :

- Struktur team
- Dokumentasi
- Pertemuan

Struktur team

Ditentukan dengan menjelaskan

- Penjelasan peranan
- Tanggung jawab
- Hubungan pelaporan

SOW sebaiknya menyediakan dasar untuk menjelaskan peranan utama, tanggung jawab, dan hubungan pelaporan. Informasi ini membantu untuk mempersiapkan struktur team, seperti untuk menghasilkan :

- diagram organisasi
- matriks tanggung jawab.

Dokumentasi

Dokumentasi adalah penting sekali, sebab user memiliki peranan penting dalam membuat dan merawat kandungan web site. Diagram arsitektur, perangkat bantu mapping, dan manual on line merupakan perangkat bantu dokumentasi teknis. Dokumentasi bisnis seperti laporan status, dan jadwal juga penting. Kedua dokumentasi baik teknis maupun bisnis, harus disimpan dalam perpustakaan yang dapat diakses untuk referensi mendatang.

Pertemuan

Terdiri dari 3 jenis :

- **Status review meeting** , dilakukan secara regular untuk mengumpulkan informasi mengenai kondisi dari pekerjaan individu.
- **Checkpoint review meeting** , dilakukan untuk mencapai milestone besar, seperti mensetup server.
- **Staff meeting** , dilakukan untuk bertukari informasi dan bertukar pengalaman bagi seluruh pihak yang terlibat

Pihak yang menghadiri pertemuan ini dapat bervariasi, tapi minimal pihak pengguna harus ada yang diundang. Ini menyebabkan mereka tidak saja merasa terlibat tetapi juga memperoleh informasi mengenai sekuriti, hak akses, dan kandungan dokumen. Hal ini akan mendorong dapat diselesaikannya proyek ini.

1.1.4 Pengawasan

Proses ini menjamin bahwa proyek Intranet efektif pembiayaanya, dan sesuai dengan yang direncanakan. Proses ini terdiri dari :

- Status collection
- Change control
- Corrective action

Status collection and Assesment

Proses ini akan mengumpulkan data tentang penyelesaian suatu pekerjaan atau pencapaian suatu milestone. Kemudian membuat penilaian mengenai perkembangan yang dilakukan. Proses ini memiliki sisi bisnis dan teknis. Sisi teknis melibatkan penilaian kualitas pekerjaan yang dilakukan misal bagaimana HTML dan CGI yang disusun. Pada sisi bisnis meliputi pada tingkatan mana pekerjaan itu dilakukan berdasarkan waktu tertentu.

Change control

Proses ini melibatkan pekerjaan mengevaluasi pelaksanaan teknis dan jadwal. Dalam pelaksanaan membutuhkan jawaban akan pertanyaan seperti :

- Apakah sebenarnya perubahan yang terjadi (misal arsitektur jaringan)
- Apa dampaknya bagi finansial, jadwal, dan kualitas sistem.
- Bagaimana penanganan perubahan tersebut, misal terhadap user dan komunitas sistem informasi.
- Bilamana perubahan tersebut akan menyebabkan suatu efek, misal setelah intranet terpasang dan berjalan.

Corrective action

Langkah ini melakukan revisi pendekatan yang dilakukan untuk mencapai tujuan proyek sesuai dengan SOW dan perencanaan. Langkah ini berkaitan sekali dengan langkah status collection and assesment, sebab langkah yang dibutuhkan misal perencanaan ulang, bergantung apakah corrective action ini perlu dilakukan secara besar atau cukup sedikit saja.

1.1.5 Penyelesaian proyek

Pada proses ini terlibat melakukan pengumpulan dan analisis data dan melakukan transisi yang baik dari proses pengembangan ke implementasi. Keluaran utama dari proses ini adalah hal yang dipelajari selama pelaksanaan proyek - *lesson learned document* . Dokumen ini mengidentifikasi apa yang dilakukan dengan baik, dan apa yang tak berhasil dilakukan. Hal itu berdasarkan data yang dikoleksi mengenai unjuk kerja proyek melalui kumpulan hasil statistik, wawancara, dan review setelah implementasi. Dokumen ini berguna bagi organisasi besar yang mungkin akan melakukan pemasangan site Intranet yang berjumlah banyak. Pengalaman yang diperoleh dari proyek pertama ini akan memberikan pandangan bagi manajer proyek untuk proyek mendatang.

Suatu hal yang penting lagi adalah bagaimana hasil dari proyek ini. Tendensi apakah yang terjadi di antara personal yang terlibat pada pengembangan proyek pada saat mendekati akhir proyek. Bila suatu proyek akan selesai biasanya anggota team menjadi menurun produktifitasnya. Oleh karena itu, sebaiknya

bila seorang anggota team telah melakukan suatu tugas berat, sebaiknya segera dibebaskan tugasnya bila memang telah tidak ada pekerjaannya lagi. Ini menyebabkan personal tersebut dapat bertugas di proyek Intranet yang lainnya lagi

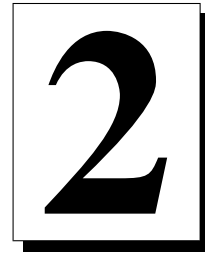
1.1.6 Leading

Tahapan ini penting sekali hanya akan terjadi bila ke lima proses sebelumnya dilakukan dengan benar. Pada tahapan ini dibutuhkan pembentukan suatu lingkungan kerja yang mendorong pihak yang terlibat, sehingga dapat tercapainya tujuan. Untuk mencapai hal tersebut, manajer proyek haruslah :

- Membuat visi yang jelas bagi proyek
- Berkomunikasi dengan efektif
- Menjaga motivasi yang tinggi
- Menjaga fokus dari visi
- Menyediakan lingkungan yang mendukung
- Mendorong penyusunan team.

Beberapa langkah tersebut sulit dilaksanakan karena biasanya manajer proyek tidak terlalu memiliki kendali dalam penggunaan sumber daya. Hal ini menjadi lebih rumit untuk proyek intranet yang melibatkan banyak pihak orang dengan keahlian terbatas, orientasi fungsi yang tak jelas. Web master dan document owner, bukanlah nama pekerjaan yang unik tapi juga membutuhkan keahlian khusus.

Suatu proyek akan dapat dilakukan dengan baik bila telah dilakukan proses engineering yang baik. Ini berlaku baik untuk pengembangan program dengan produk jadi, atau dengan kontraktor atau juga dengan team sendiri. Akan lebih baik menghabiskan waktu lebih lama untuk melakukan disain dan penataan awal yang baik, daripada terburu-buru melakukan implementasi. Sehingga sudah sewajarnya dilakukan standardisasi, dan penggunaan dokumentasi yang baik. Biasanya suatu team pengembang sistem informasi cenderung untuk meninggalkan metodologi standard dengan alasan keterbatasan waktu. Rapid Application Development (RAD) bukanlah merupakan suatu alasan untuk tidak menggunakan teknik-teknik disain yang baik.



Dokumen perancangan sistem

Aktifitas Perencanaan Pengembangan perangkat lunak :

- Memilih suatu model untuk proses pengembangan
- Mengontrak dan menyusun team pembuat software
- Membeli atau menyewa perangkat keras/lunak yang dibutuhkan
- Memilih produk berpotensi berdasarkan pengalaman keorganisasian, sumber daya, dan tujuan.
- Mengevaluasi informasi pasar mengenai viabilitas produk
- Mengestimasi biaya, jadwal, resiko dan harga dari produk
- Memilih bentuk laporan dan cara pengukuran perkembangan proyek
- Memilih notasi untuk spesifikasi dan perancangan
- Memilih bahasa pemrograman
- Meletakkan standard untuk dokumentasi, coding, verifikasi dan pengujian
- Memilih mekanisme pengaturan konfigurasi
- Menyiapkan bahan dan personil untuk instalasi

Beberapa dokumen yang biasa digunakan adalah :

2.1 Project plan

Suatu project plan (perencanaan proyek) berisi :

1. Pengantar, berisi :

- Deskripsi permasalahan
- Deskripsi lingkungan masalah
- Tujuan client, dan organisasi serta sistem.
- Solusi yang diajukan dan ruang lingkupnya

2. Proposal

- Fungsi yang diberikan pada solusi yang diajukan
- Strategi umum untuk mengembangkan solusi
- Peran pengguna dan perangkat keras pada solusi tersebut
- Keuntungan dan kerugian solusi tersebut

3. Keterbatasan sistem (*constraint*)

- Prioritas kustomer
- Profil pengguna
- Usia pengharapan dari produk
- Pra-syarat keandalan (reliabilitas)
- Pra-syarat kinerja
- Lingkungan perangkat keras dan antar muka yang telah ada
- Pengembangan mendatang dari produk
- Pra-syarat, bahasa pemrograman untuk implementasi (jika ada)
- Pra-syarat pelatihan, instalasi dan dokumentasi.
- Ketersediaan pada lingkungan pengguna
- Solusi alternatif
- Studi feasibilitas

4. Estimasi

- Jadwal
- Staf dan organisasi
- Budget
- Analisis Cost/Benefit
- Analisa resiko
- Dokumen yang diberikan
- Perangkat lunak yang dibutuhkan
- Fasilitas dan perangkat keras yang dibutuhkan

5. Prosedur

- Model proses
- Metodologi dan notasi
- Standardisasi dan jaminan kualitas
- Accountability monitoring
- Kendali produk
- Data pengujian dan sumber data
- Kriteria akseptansi dan metoda pembayaran

6. Referensi

- Dokumentasi yang digunakan dalam pengembangan
- Kamus istilah
- Kontrak yang diusulkan (jika ada)

2.2 Spesifikasi disain

Dokumen ini pada dasarnya menerangkan tentang kebutuhan sistem yang akan dibuat. Beberapa paradigma perancangan akan menentukan model disain dan notasi. Beberapa pendekatan disain adalah :

- Access-oriented design
- Data-structure-oriented design
- Data flow design
- Functional design
- Imperative design
- Object-oriented design
- Parallel design
- Real-time design
- Rules-oriented design
- User centered design

Suatu dokumen spesifikasi disain terdiri dari :

1. Pendahuluan

- Garis besar permasalahan
- Lingkungan aplikasi dan karakteristik pengguna

- Notasi yang digunakan dalam disain
- Tujuan proyek

2. Spesifikasi secara singkat

- Fungsi perangkat lunak
- Teknik yang digunakan untuk menyelesaikan masalah ini terutama ketika disainer tidak memiliki pengetahuan khusus
- Kinerja yang harus dicapai
- Deskripsi data
- Hubungan data
- Prioritas implementasi
- Spesifikasi real time
- Spesifikasi interaksi manusia dan mesin yang digunakan.
- Batasan
- Eksepsi
- Modifikasi dan perawatan yang diprediksi

3. Disain arsitektur

- Modul hirarki dan diagram interface
- Deskripsi fungsi dan data
- Spesifikasi interface

4. Disain secara detail

- Untuk tiap modul :
 - Deskripsi modul dan spesifikasi interface
 - Deskripsi proses
 - Definisi struktur data
 - Pra-syarat inisialisasi
 - Spesifikasi penanganan eksepsi
 - Alternatif disain, untuk tiap disain yang ditolak disertakan keterangan alasan penolakan serta kondisi yang menyebabkan disain yang terpilih.

5. Referensi

- Dokumentasi yang digunakan untuk mengembangkan disain
- Daftar terminologi



Contoh model dokumentasi

Berikut ini diberikan suatu contoh dokumentasi perancangan suatu sistem komputer yang berisi :

- Bagian 1. Kebutuhan pengguna
- Bagian 2. Spesifikasi
- Bagian 3. Disain
- Bagian 4. Implementasi dan pemilihan teknologi
- Bagian 5. Pengujian
- Bagian 6. Aplikasi (bila perlu)
- Lampiran

3.1 Kebutuhan pengguna (*user requirement*)

Bagian ini menjelaskan kebutuhan dari sistem baik, dari sis pengguna, fungsi maupun teknologi. Bagian ini terdiri dari :

3.1.1 Definisi Kebutuhan (*Requirement definition*)

Bagian ini mendefinisikan kebutuhan dan hal-hal yang berkaitan dengan kebutuhan akan sistem secara menyeluruh. Biasanya dalam kasus kondisi sebenarnya, dokumen ini harus disetujui oleh pemberi kerja dan pelaksana kerja. Bagian yang dibahas dalam dokumen ini adalah :

- **Purposeful requirement.**
Menjelaskan mengapa kebutuhan itu perlu dipenuhi

- **Functional requirement.**
Apakah fungsi sebenarnya yang dibutuhkan oleh user dari sistem ini.
- **Nonfunctional requirement.**
Dan kebutuhan sistem yang tidak berkaitan dengan fungsinya yang melibatkan seperti, bisa dimodifikasi, testing, dan portabilitas
- **User Profile.**
Menerangkan tentang pengguna dari sistem ini, terutama yang berkaitan dengan tingkat pengenalan terhadap teknologi yang akan diterapkan.

3.1.2 Analisis Kebutuhan (*Requirement Analysis*)

Bagian ini akan menganalisis kebutuhan yang dijabarkan pada bagian sebelumnya serta kaitannya dengan hal-hal yang mempengaruhi pelaksanaan ataupun pemilihan solusi.

- **Requirement prioritisation.**
Menjabarkan kebutuhan mana yang terpenting dari kebutuhan-kebutuhan yang timbul untuk sistem yang akan dibuat tersebut.
- **Constraint and Risk Analysis.**
Memahami halangan dan resiko yang ada untuk mengimplementasikan model dan mengaplikasikan sistem ini.
- **Trade-off analysis.**
Mengetahui hal-hal yang saling bertentangan dalam perancangan, pengimplementasian serta pengaplikasian dari solusi.

3.1.3 Model Kebutuhan (*Requirement model*),

Bagian ini menyusun model kebutuhan tersebut menjadi suatu bentuk model kebutuhan. Disusun a hierarchical (functional) model dari prioritas, risk functional, non-functional requirement dan suatu prototype yang interactive. Model ini digunakan untuk mengurangi ketidak pastian kebutuhan sistem, sehingga tercapai kesepakatan antara pemberi kerja dan pelaksana kerja..

- Functional model
- Exploratory prototype
- Requirement trace
- The conceptual model

Joint Requirement Planning Diagram, dapat digunakan untuk mendapatkan user requirement.

3.2 Spesifikasi

Pada bagian ini dijabarkan spesifikasi detail dari sistem yang harus dibuat, spesifikasi ini meliputi hal-hal berikut ini :

3.2.1 Spesifikasi siklus operasi sistem (*System Operating Cycle Specification*)

Dalam bagian ini dijelaskan siklus penggunaan sistem ini, tujuan untuk mendapatkan spesifikasi teknis terutama akan berguna pada bagian implementasi. Sistem fault tolerant 24 jam akan berbeda dengan sistem yang beroperasi 1 jam sehari. Begitu juga siklus kerja sistem, atau langkah kerja satu demi satu dispesifikasikan.

3.2.2 Spesifikasi fungsional

Pada bagian ini dijabarkan fungsi dari sistem yang akan dibuat.

- **Essential Capabilities**
Dijabarkan fungsi utama dari sistem. Ini merupakan fungsi minimal yang harus dipenuhi oleh sistem yang dibuat.
- **Additional Capabilities**
Dideskripsikan fungsi tambahan yang timbul karena dipenuhi dan digunakan sistem ini.
- **Future Capabilities**
Menjelaskan fungsi pada masa datang dari sistem ini. Penjelasan ini erat kaitannya dengan *marketing strategy*, *technological*, dan *Sales*

3.2.3 Komponen sistem

Pada bagian ini dijabarkan komponen-komponen yang dibutuhkan untuk sistem secara keseluruhan. Termasuk software, hardware, user, dan organisasi penunjang. Untuk mempermudah penjabaran pembagian tugas antar komponen dapat dilakukan *process decomposition* dan penjabaran *relationship* antar komponen.

3.2.4 Spesifikasi kinerja

Pada bagian ini dijelaskan performansi / unjuk kerja yang ingin dicapai serta kemungkinan keterbatasan di dalam penggunaan sistem. Setiap kebutuhan diusahakan dispesifikasikan dengan jelas, termasuk karakteristik elektrik dan fisis.

- **Characteristics and Constraints**
Dijabarkan karakteristik tiap komponen pendukung sistem, termasuk karakteristik fisis, elektrik, maupun karakteristik perangkat lunak (misal

user friendly, interaktif). Juga dijabarkan keterbatasan di dalam penggunaan sistem (misal pengguna harus menggunakan kaki sebagai masukan).

- **Physical characteristics**

Dijabarkan bentuk luar dari sistem secara keseluruhan. Misal ukuran berat dan lain-lain.

- **Environmental characteristics**

Lingkungan tempat kerja dari sistem ini di

- **Human Factors**

Dijabarkan pengaruh manusia di dalam operasi sistem, baik sebagai penentu operasi, misal dalam decision making system, atau supervised control system. Juga pengaruh manusia dalam menimbulkan ketidakakuratan atau kesalahan sistem.

Untuk mempermudah proses spesifikasi dapat digunakan *prototyping*, contoh I/O screen, contoh keluaran. Proses pelaksanaan spesifikasi ini dalam pelaksanaan proyek sesungguhnya biasanya menggunakan metoda formal *Joint Application Design* (JAD).

3.3 Disain

Pada bagian ini diterangkan metoda dari solusi yang digunakan untuk memenuhi kebutuhan yang telah dispesifikasikan pada bagian terdahulu. Yang dijelaskan dalam bagian ini adalah langkah perancangan dari solusi yang ditawarkan. Sedapat mungkin harus dipisahkan antara perancangan dan implementasi. Pada perancangan diusahakan sedapat mungkin yang dilakukan adalah memodelkan solusi secara logika atau secara algoritmis. Tanpa terkait erat dengan teknologi yang digunakan dalam proses implementasi dari model. Keterkaitan implementasi hanyalah menjadi constraint dari model bukan menjadi dasar disain. Langkah-langkah disain akan sangat bergantung pada model dari sistem yang dibuat. Pada dasarnya akan meliputi :

3.3.1 Disain sistem utama

- **Diagram blok**

Sebagai langkah pertama adalah menggambar blok pembangun sistem secara keseluruhan juga digambarkan interface antara kedua bagian tersebut. Blok diagram terdiri dari 2 bagian :

- Dunia di luar sistem
- Sistem

- **Flow of control** ,Dapat menggunakan Algoritmik State Machine (ASM). Yang menerangkan kondisi-kondisi yang mungkin =dari sistem serta transisi dari satu kondisi ke kondisi lainnya. Diturunkan dari operasi sistem.

- **Representation of Data Flow**
Menggambarkan aliran data serta transformasi yang dialami data tersebut dalam sistem
- **Decomposition into functions**
Membagi-bagi fungsi secara keseluruhan menjadi sub sistem dengan sub fungsinya. Dapat digunakan *Tree Diagram*.
- **Relationship among function**
Dijelaskan hubungan antar sub sistem dengan kaitannya terhadap sub fungsinya.
- **Module specification**

Dalam memilih diagram atau notasi untuk mendisain perlu diperhatikan beberapa point :

- **Suatu alat bantu untuk berfikir secara jelas**
Suatu diagram yang baik akan membantu orang untuk memahami ide yang kompleks. Suatu diagram notasi harus direncanakan untuk mempermudah cara fikir dan komunikasi dan untuk berfikir yang berbantuan komputer. Diagram merupakan alat bantu
- **Mudah dimengerti**
Suatu diagram harus digunakan bila memiliki pengertian yang jtelah dikenal. Harus dihindari simbol yang sulit dijelaskan.
- **Sebagai alat bantu untuk komunikasi dengan end-user**
End user harus dapat memahami disain dan memberikan masukan kepada disainer. Jadi penggunaan notasi yang membingungkan end-user sebaiknya dihindari.

3.4 Implementasi dan pemilihan teknologi

Pada bagian ini dijelaskan metoda dan peralatan yang digunakan untuk mengimplementasikan solusi yang telah diajukan dalam bagian disain. Sebaiknya alasan pemilihan teknologi yang digunakan haruslah dijabarkan pada bagian ini. Misal alasan pemilihan suatu jenis komponen perlu diberikan dengan jelas dalam bagian ini, misal mengapa memilih CMOS atau TTL.

Dalam bagian implementasi diterangkan skema atau diagram yang digunakan, baik elektronis, fisis, ataupun keterangan program dengan menggunakan metoda diagram yang sesuai. Misal untuk rangkaian elektronis menggunakan skema elektronis, sedangkan untuk program dengan flow chart, dan lain lain.

Sebelum dilakukan pemilihan teknologi atau level alat yang digunakan maka harus dilakukan estimasi terhadap beberapa hal :

- Estimasi waktu untuk mengembangkan sistem

- Estimasi panjangnya program
- Estimasi kebutuhan memori
- Estimasi kecepatan eksekusi

Juga harus dipertimbangkan *pembagian antara software dan hardware* .

3.5 Pengujian (testing)

Bagian ini menunjukkan kerja dari sistem baik untuk masukan yang bersifat normal, ataupun untuk masukan yang di luar ambang normal. Setiap pengujian dilakukan dokumentasi sebagai bukti otentik kemampuan sistem. Pada dasarnya disamping testing yang bersifat testing dalam kondisi operasi, akan baik pula bila dilakukan testing berikut ini :

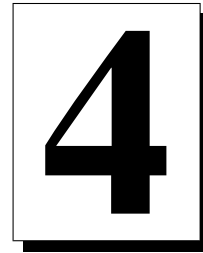
- **Recovery testing.**
Pengujian dilakukan untuk mengetahui kemampuan sistem, untuk mengembalikan ke kondisi normal setelah suatu masukan atau kondisi di luar dari yang dispesifikasikan. Untuk sistem-sistem yang bersifat fault-tolerant, jenis pengujian ini merupakan suatu kewajiban.
- **Stress testing.**
Pengujian ini dilakukan untuk mengetahui kemampuan sistem dalam menangani beban kerja yang berat, sangat baik untuk mengetahui kemampuan maksimal dari sistem.
- **Security testing.**
Testing ini dilakukan untuk jenis-jenis aplikasi yang berkaitan dengan keamanan sistem, baik dari pengguna yang melakukan kesalahan tidak sengaja, ataupun sengaja.

Dapat digunakan beberapa metodologi testing

- Use-Case
- White Box
- Black Box
- Loop testing

Dapat digunakan beberapa satuan untuk menunjukkan hasil kerja sistem, bila dalam perangkat keras telah jelas besaran yang digunakan misal : frekuensi response, slew rate dll. Untuk perangkat lunak dapat digunakan : software metric, cyclomatic complexity. Pada dasarnya suatu testing akan melakukan :

- **Verifikasi** .
Menjamin bahwa sistem benar-benar bekerja sesuai fungsinya.
- **Validasi** .
Menjamin bahwa sistem benar-benar memenuhi keinginan pemakai.



Dokumentasi tambahan

4.1 Manual penggunaan

Bagian ini berisi manual yang memberikan petunjuk cara pengoperasian alat. Secara singkat diberikan petunjuk pemakaian dari program tersebut. Terutama tentang menu-menu dari program ini. Terdiri dari :

1. Pengantar
 - Tujuan dari produk
 - Lingkungan operasi
 - Fungsi secara umum
 - Fitur khusus
 - Keterbatasan
 - Keterangan dan notasi dokumen ini
2. Installasi
 - Persyaratan minimal sistem yang dibutuhkan
 - Menyalin dan memback-up
 - Proses instalasi
 - Menkonfigurasi /kustomisasi produk
3. Tutorial
 - Penjelasan langkah-demi langkah dengan contoh
 - Penjelasan tiap contoh
 - Pengembangan dari contoh dasar

- Penggunaan Help online dan petunjuk penggunaan (manual)

4. Instruksi detail

- Keluaran dari produk
- Masukan untuk produk
- Pengoperasian produk
- Penanganan error
- Fungsi khusus

5. Detail teknis

- Prinsip dari operasi
- Fitur lanjutan
- Algoritma utama yang digunakan
- Struktur data utama
- Modifikasi produk
- Cara memperoleh dukungan teknis dan informasi lanjutan

4.2 Maintenance Manual

Bagian ini menerangkan tata cara perawatan dan pengelolaan sistem yang baik, agar sistem dapat beroperasi dengan aman dan memberikan fungsinya sebaik mungkin. Dibagi menjadi :

- **Maintenance Manual.**

Sub bagian ini hanya memberikan tentang tata cara perawatan baik secara elektrik, fisis maupun perawatan perangkat lunak yang harus dilakukan.

- **Trouble shooting Manual.**

Sub bagian ini memberikan petunjuk tentang yang harus dilakukan oleh pengguna sistem bila terjadi kerusakan, atau ketidak normalan kerja sistem. Tingkat kerusakan yang ditulis biasanya hanyalah sampai pada level yang ringan dan tak perlu penanganan secara khusus.

4.3 Dokumentasi testing.

Untuk menunjukkan performansi dari sub sistem dan sistem dijabarkan pada bagian ini. Testing bisa dilakukan untuk :

- **Sub rutin yang penting**

Ini dilakukan dengan cara memasukkan nilai input ke sub rutin tersebut dan hasil dari sub rutin tersebut di-print. Hal ini dilakukan untuk sub-rutin yang pokok. Nilai input yang digunakan adalah nilai input yang menunjukkan kerja dari sub rutin tersebut.

- **Program secara keseluruhan**

Ini dilakukan dengan memberikan suatu nilai input tertentu, yang sudah diketahui outputnya. Dengan membandingkan hasil program dan acuan tersebut bisa dilihat hasil kerja dari program tersebut.

4.4 Dokumentasi source code

Untuk dokumentasi dari satu program harap dilakukan :

- Penamaan variable, constanta, procedure, function yang jelas dan menunjukkan fungsinya. Usahakan seragam dan jelas serta konsisten.
- Pada header setiap prosedur harus dilengkapi dengan keterangan :
 - Fungsi dari prosedur, dan algoritma yang digunakan (jika spesifik)
 - Variable lokal masukan, dan variable lokal keluaran
 - Variable global yang digunakan, variable global yang dipengaruhi.
- Pada header dari program utama diberikan :
 - Nama penulis program
 - Editor
 - Compiler dan Library yang digunakan
 - Versi dan upgrade history
 - Tanggal pembuatan software
 - Deskripsi singkat tentang software
- Sedangkan pada tiap modul diberikan informasi
 - Nama modul :
 - Fungsi :
 - Parameter interface dan modus :
 - Preassertion :
 - Postassertion :
 - Dampak global dan sampingan :
 - Exception :
 - Prasyarat perangkat keras dan sistem operasi :
 - Catatan pembuatan dan modifikasi :
 - Algoritma :
 - Struktur data utama :
 - Called by:
 - Calls :

4.5 Dokumentasi testing

Biasanya terdiri dari

- Kebutuhan (menyatakan ulang dari dokumentasi spesifikasi)
- Metodologi verifikasi disain, jadual dan pihak yang bertanggung-jawab
- Metodologi verifikasi kode, jadual, pihak yang bertanggungjawab
- Response data pengujian, jadual dan pihak yang bertanggungjawab
- Rencana pengujian
- Fitur dan sisi yang akan diujia
- Personal yang bertanggungjawab serta jadual
- Perangkat bantu atau program bantu yang dibutuhkan
- Data pengujian dan instruksi pengujain
- Hasil pengujian yang diharapkan
- Hasil pengujian sesungguhnya, serta analisis (jika ada)