

Nama : Nofa Arsyadana SS

NIM : I0720055

Tugas Pengambilan Sistem Keputusan Pertemuan 5

DEVELOPMENT OF DECISION SUPPORT SYSTEM

1. Introduction

Dalam literatur Sistem Pendukung Keputusan (DSS), para ahli meresepkan beragam pendekatan atau metodologi untuk merancang dan mengembangkan DSS. Namun, mereka tidak sepakat mengenai metodologi apa yang paling tepat untuk membangun hal-hal yang berbeda jenis DSS. Jika manajer dan analis DSS memahami berbagai metode, mereka dapat membuat pilihan yang lebih terinformasi dan lebih baik ketika membangun atau membeli DSS tertentu. Secara umum, apa yang disebut “pendekatan berorientasi pada keputusan” tampaknya paling tepat digunakan proyek Sistem Pendukung Keputusan. Setelah meninjau desain dan pengembangan masalah, diagnosis berorientasi keputusan, dan studi kelayakan, bab ini mengulas tiga pendekatan alternatif untuk mengembangkan DSS. Karena ruang lingkup DSS adalah berkembang, dan karena alat-alat pembangunan berubah dengan cepat, hal tersebut dirasakan keuntungan dari tiga pendekatan pembangunan alternatif telah menjadi agak kontroversial. Misalnya, siklus hidup yang sangat terstruktur Pendekatan pembangunan baru-baru ini menjadi populer di kalangan beberapa konsultan mengembangkan DSS di seluruh perusahaan.

2. Overview Of Design And Development Issues

Bagaimana merencanakan dan melaksanakan sebuah DSS baru? Apa artinya merancang sebuah DSS? Bagaimana cara mengembangkan sebuah DSS? Siapa yang mengembangkan DSS baru? Kapan sebuah perusahaan sebaiknya membangun sebuah DSS dan kapan sebaiknya membeli paket DSS? Baik manajer maupun para profesional Sistem Informasi Manajemen (MIS) perlu menjelajahi pertanyaan-pertanyaan ini. Sebuah perusahaan tidak akan mendapatkan keuntungan dari ide brilian untuk DSS sampai sistem baru tersebut dibangun dan berhasil diimplementasikan.

Banyak profesional Sistem Informasi (SI) mengembangkan, mengubah, dan menyesuaikan perangkat lunak untuk mendukung pengambilan keputusan. Mereka bekerja dalam berbagai konteks bisnis dan organisasi, termasuk di perusahaan-

perusahaan khusus pengembangan perangkat lunak DSS. Vendor perangkat lunak DSS menjual beragam produk dan menyediakan layanan pengembangan DSS. Misalnya, Comshare (www.comshare.com) dan Cognos (www.cognos.com) keduanya memasarkan produk bisnis intelligence dan perencanaan serta kontrol manajemen.

Desain dan pengembangan adalah topik yang penting karena DSS melayani berbagai fungsi yang berbeda dan sangat beragam dalam hal perangkat lunak yang digunakan untuk pengembangannya. Memilih pendekatan atau metodologi yang sesuai untuk membangun DSS telah menjadi topik populer dan kontroversial dalam literatur Sistem Informasi (SI). Banyak firma konsultasi berfokus pada penggunaan metodologi pengembangan yang mereka klaim paling efektif. Kita dapat mendefinisikan metodologi sebagai serangkaian praktik dan prosedur terorganisir yang digunakan oleh pengembang. Meskipun ada banyak perbedaan dalam metodologi dan terminologi, rekomendasi dalam literatur SI secara umum telah mengikuti tiga jalur konseptual yang berbeda.

Sebuah kelompok ahli MIS dan DSS mengembangkan rekomendasi mereka untuk membangun DSS dalam konteks literatur analisis dan desain sistem tradisional (cf., Thierauff, 1982). Kelompok kedua telah meresepkan dan menjelaskan pendekatan berulang, prototyping, atau pendekatan "quick-hit" untuk merancang dan mengembangkan DSS (cf., Sprague dan Carlson, 1982). Beberapa penulis merujuk kepada kedua jenis pendekatan tersebut tanpa menjelaskan dengan jelas kelebihan dan kekurangan atau kondisi yang mendukung suatu pendekatan tertentu atau kombinasi pendekatan. Pendekatan ketiga dalam membangun DSS, yang disebut pengembangan oleh pengguna akhir, adalah membiarkan manajer mengembangkan DSS pribadi mereka sendiri. Secara umum, literatur yang meresepsi DSS mengenai desain dan pengembangan didasarkan pada pengalaman pribadi, studi kasus, literatur pengembangan SI umum, dan berbagai "kisah perang" DSS dari pengembang. Penelitian empiris yang sangat sedikit telah dilakukan mengenai metodologi desain dan pengembangan.

Karena masalah desain dan pengembangan, beberapa DSS yang sangat inovatif dan berpotensi berguna telah gagal. Masalahnya sering kali adalah bahwa DSS dirancang dan dikembangkan dari sudut pandang pemrogram dan pengembang daripada dari sudut pandang manajer dan pengguna. Urutan perintah atau ikon mungkin jelas bagi pemrogram, tetapi mungkin sama sekali tidak dikenal dan membingungkan bagi pengguna DSS. Dari sudut pandang preskriptif, DSS yang efektif harus

berorientasi pada pengguna. Isu kunci adalah proses desain dan pengembangan apa dan prosedur mana yang dapat meningkatkan kemungkinan bahwa DSS yang dapat digunakan dan efektif akan diciptakan dan dibangun.

Membangun DSS seringkali sangat mahal. Jadi, penting untuk menyelidiki pendekatan desain dan pengembangan alternatif. Kami ingin memilih pendekatan yang meningkatkan peluang DSS akan digunakan dan akan mencapai tujuannya. Kami perlu mengingat bahwa DSS dirancang dan dikembangkan untuk membantu orang membuat keputusan yang lebih baik dan lebih efektif daripada yang bisa mereka lakukan tanpa bantuan komputer. Membangun jenis DSS apapun sulit karena orang sangat bervariasi dalam hal kepribadian, pengetahuan, kemampuan, preferensi, pekerjaan yang mereka pegang, dan keputusan yang perlu mereka buat. Selain itu, DSS seringkali harus memenuhi berbagai persyaratan yang beragam. Beragamnya persyaratan yang berbeda ini telah mengarah pada desain dan pengembangan beragam kemampuan dan sistem DSS.

3. Decision-Oriented Diagnosis

Meningkatkan efektivitas pengambilan keputusan melalui perubahan dalam cara keputusan dibuat harus menjadi tujuan utama untuk setiap proyek DSS (lihat Stabell 1983). Stabell mengusulkan pendekatan desain berorientasi pada keputusan untuk DSS. Dia berargumen bahwa predesain deskripsi dan diagnosis tentang pengambilan keputusan adalah kunci untuk memastikan pendekatan berorientasi pada keputusan dalam pengembangan DSS.

Diagnosis tentang pengambilan keputusan saat ini dan spesifikasi perubahan dalam proses pengambilan keputusan adalah aktivitas yang memberikan masukan kunci untuk desain DSS. Diagnosis adalah identifikasi masalah atau peluang untuk perbaikan dalam perilaku pengambilan keputusan saat ini. Diagnosis melibatkan penentuan bagaimana keputusan saat ini dibuat, menentukan bagaimana keputusan seharusnya dibuat, dan memahami mengapa keputusan tidak dibuat seperti seharusnya. Spesifikasi perubahan dalam proses pengambilan keputusan melibatkan pemilihan perbaikan tertentu dalam perilaku pengambilan keputusan yang harus dicapai. Pernyataan-pernyataan perbaikan ini memberikan tujuan untuk pengembangan DSS.

Menurut Stabell, diagnosis masalah dalam proses pengambilan keputusan melibatkan tiga aktivitas berikut:

- a. Mengumpulkan data tentang pengambilan keputusan saat ini menggunakan teknik seperti wawancara, observasi, kuesioner, dan catatan historis;

- b. Membuat deskripsi yang koheren tentang proses pengambilan keputusan saat ini
- c. Menentukan norma tentang bagaimana keputusan seharusnya dibuat.

Aktivitas-aktivitas ini saling terkait dan memberikan umpan balik untuk analisis DSS. Dalam banyak proyek pengembangan DSS, tidak memungkinkan untuk melakukan diagnosis pengambilan keputusan dalam skala penuh. Studi singkat seringkali diperlukan karena pertimbangan biaya, akses terbatas ke para manajer, atau kendala organisasi lainnya. Oleh karena itu, analisis DSS harus mengembangkan kemampuan untuk menghasilkan diagnosis setelah analisis yang terbatas dari situasi pengambilan keputusan.

Aktivitas diagnostik terkait adalah melakukan Audit Proses Keputusan. Secara umum, sangat berguna untuk mengaudit proses pengambilan keputusan operasional dan manajerial. Audit dapat menjadi langkah awal dalam mengidentifikasi peluang untuk merancang ulang proses bisnis dan menyertakan bantuan pengambilan keputusan baru dan DSS dalam proses bisnis. Dalam beberapa situasi, audit dapat menyarankan perubahan dalam teknologi pengambilan keputusan yang dapat meningkatkan kinerja dan mengurangi biaya. Ketika audit selesai, pertanyaan utamanya haruslah bagaimana kita bisa melakukan lebih baik dan perubahan apa yang harus memiliki prioritas tertinggi. Tabel 4.1 mengidentifikasi lima langkah dalam Audit Proses Keputusan. Membuat Diagram Aliran Data (Data Flow Diagram - DFD) adalah langkah penting dalam Audit Proses Keputusan. DFD menggambarkan proses bisnis dan aliran data melalui proses tersebut secara grafis. Untuk membuat DFD, analisis DSS memecah proses yang sedang diteliti menjadi langkah-langkah kecil, tindakan, atau peristiwa. Langkah-langkah ini dapat terjadi secara berurutan atau sekaligus. DFD sangat berguna untuk meningkatkan komunikasi antara analisis DSS dan para manajer.

	Decision Process Audit Plan
Step 1.	Define the decisions, decision processes and related business processes that will be audited. Define the authority of the auditor, purpose of the audit, scope of the audit, timing of the audit, and resources required to perform the audit. Identify a primary contact.
Step 2.	Examine the formal design of the process. Diagram the process using Data Flow Diagrams and specify participants, data, criteria, etc.
Step 3.	Examine the actual use of the decision process. Observe the process. Interview decision makers and collect data. Is the process implemented and used as intended?
Step 4.	Assess performance of the actual decision process. What works? Can cycle time be reduced? Are decisions appropriate? Timely? Cost effective? Is the process producing value in meeting business objectives? If not, why?
Step 5.	Reporting and recommendations. Summarize steps 1-4 in a written report. Discuss what is working well and what needs to be improved. Develop recommendations for improving the process. Hold an exit meeting with decision makers.

Table 4.1. A Decision Process Audit Plan.

Sebuah audit juga seharusnya fokus pada mengidentifikasi apa yang diasumsikan oleh para pengambil keputusan dalam situasi pengambilan keputusan dan apa yang didefinisikan oleh para pengambil keputusan sebagai kisaran tindakan perbaikan yang tersedia. Mengidentifikasi asumsi dan tindakan adalah khususnya penting jika membangun DSS berbasis model adalah kemungkinan. Menilai kinerja proses pengambilan keputusan aktual adalah tugas penting dalam audit. Kita perlu menentukan tugas mana yang efektif. Dapatkah waktu siklus dikurangi? Apakah keputusan tepat? Tepat waktu? dan Efektif biaya? Beberapa proses seperti perencanaan bisnis memerlukan akses data yang lebih baik dan analisis untuk meningkatkan bisnis intelligence. Rockart (1979) mengidentifikasi pendekatan untuk mendefinisikan kebutuhan data dalam pengambilan keputusan yang sesuai untuk DSS berbasis data dan khususnya Sistem Informasi Eksekutif (Executive Information Systems - EIS). Metode Desain Faktor Kesuksesan Kritis (Critical Success Factors - CSF) yang digagas oleh Rockart berfokus pada manajer individu dan pada kebutuhan informasi keras dan lunak saat ini dari setiap manajer. Analisis CSF dapat bermanfaat dalam mengidentifikasi "sejumlah terbatas area di mana hasil, jika mereka memuaskan, akan menjamin kinerja kompetitif yang sukses bagi organisasi." Jika tujuan organisasi akan dicapai, maka area-area utama ini dalam aktivitas—biasanya tiga hingga enam faktor—akan memerlukan perhatian yang hati-hati dan konsisten dari manajemen.

Diagnosis yang baik sulit, tetapi diagnosis DSS melibatkan keterampilan yang dapat dikembangkan dan ditingkatkan. Baik manajer maupun staf MIS perlu bekerja

sama dalam menyelesaikan tugas diagnosis. Apakah diagnosis selalu memberikan informasi yang cukup untuk menentukan DSS? Dalam kebanyakan kasus, diagnosis memang memberikan informasi yang cukup untuk menentukan beberapa desain alternatif. Desain DSS biasanya melibatkan sejumlah kompromi yang sulit. Kompromi pertama adalah apakah DSS harus mendukung proses yang ada dan proses baru yang telah ditentukan. Ada juga kompromi dalam tingkat kemampuan DSS dan cakupan proses yang DSS dirancang untuk mendukung. Dalam kebanyakan kasus, versi awal DSS fokus pada kemampuan yang luas untuk proses cakupan yang sempit atau pada beberapa kemampuan untuk proses cakupan yang luas.

4. Prepare A Feasibility Study

Diagnosis pengambilan keputusan seharusnya diikuti oleh aktivitas inisiasi tambahan dan diagnosis serta penyusunan studi kelayakan mengenai prospek teknis dan ekonomi yang terkait dengan pengembangan DSS. Studi ini seharusnya dilakukan sebelum benar-benar mengalokasikan sumber daya untuk mengembangkan DSS yang diusulkan. Apa yang seharusnya disertakan dalam studi kelayakan DSS? Ini adalah pertanyaan umum. Rangkuman untuk laporan studi kelayakan yang lengkap termasuk dalam Tabel 4.2. Rangkuman ini memiliki 15 bagian mengenai topik seperti cakupan DSS dan pengguna target, dampak DSS yang diantisipasi, manfaat, risiko, dan faktor mitigasi. Studi dan laporan yang lebih pendek dan kurang komprehensif biasanya disiapkan untuk proyek DSS dengan cakupan yang lebih kecil.

I.	EXECUTIVE SUMMARY
A.	Key Business Needs
B.	Issues
C.	Solutions
D.	Benefits and Costs
E.	Critical Success Factors
F.	Project Management
II.	INTRODUCTION
A.	Background and Definitions
B.	Key Questions
1.	Site Readiness: To what extent is the company ready for and interested in implementing a new Decision Support System? See Appendix II.
2.	Technical Feasibility: Is it possible to develop or adapt software to perform the proposed types of analyses.
3.	Financial Feasibility: What are the projected costs of implementing the DSS, and do potential benefits justify these costs?
C.	Feasibility Study Approach
III.	BACKGROUND NEEDS AND ASSESSMENT
A.	Goals
B.	Constraints
C.	Related Projects
D.	Business Decision Support Needs
E.	Decision Support Diagnosis
IV.	OBJECTIVES
V.	DSS SCOPE AND TARGET USERS
A.	Scope and Decision Process Definition
B.	Scope Recommendation
C.	Scope Issues
VI.	ANTICIPATED DSS IMPACTS
VII.	PROPOSED SOLUTION
A.	System Integration Issues
B.	Major Functions Provided
C.	Technology Tools/Infrastructure Used
D.	New Organizational Structure and Processes
VIII.	MAJOR ALTERNATIVES
IX.	CONFORMITY WITH CURRENT IS/IT PLAN
X.	PROJECT MANAGEMENT AND ORGANIZATION ISSUES
XI.	ESTIMATED TIME FRAME AND WORKPLAN
XII.	INCREMENTAL COSTS AND BENEFITS
XIII.	RISKS AND MITIGATING FACTORS
XIV.	DRAFT CONCEPTUAL DESIGN

Table 4.2 DSS Feasibility Study Outline.

Studi kelayakan Sistem Pendukung Keputusan menguji konsekuensi dan dampak dari suatu proyek yang diusulkan. Studi kelayakan dirangkum dalam laporan formal atau dokumen. Studi ini mengatasi masalah termasuk manfaat, biaya, efektivitas proyek, alternatif yang dipertimbangkan, analisis alternatif, pendapat pengguna potensial, dan faktor-faktor lainnya. Analisis kelayakan ini adalah cara untuk mengeksplorasi faktor-faktor dan risiko yang memengaruhi potensi pengembangan dan implementasi yang sukses dari DSS. Upaya pengembangan sistem informasi berukuran besar biasanya melibatkan studi kelayakan sebagai titik kontrol utama yang memberikan informasi kritis tentang apakah mungkin untuk mengembangkan sebuah sistem, mengingat tujuan dan kendala proyek. Laporan ini seharusnya dirancang untuk memberikan informasi penting tentang beragam isu yang kemungkinan akan memengaruhi keberhasilan dan oleh karena itu harus dipertimbangkan dalam keputusan tentang apakah dan bagaimana melanjutkan dengan upaya pengembangan DSS. Setelah menyelesaikan studi kelayakan, keputusan seringkali dibuat antara membeli paket aplikasi dan pengembangan dalam perusahaan. Secara umum, aplikasi DSS berpaket cukup serbaguna dan biasanya lebih murah untuk diimplementasikan daripada pengembangan dalam perusahaan. Solusi berpaket juga seringkali lebih cepat untuk

diimplementasikan. Selain itu, sebuah DSS berpaket dapat mengurangi masalah politis jika sebuah proyek DSS gagal. Namun, masalah yang terkait dengan membeli DSS berpaket seharusnya tidak diabaikan. Sebuah paket mungkin tidak "cocok" dengan kebutuhan yang telah diidentifikasi, dan pesaing juga dapat membeli paket tersebut. Menggunakan aplikasi berpaket kurang mungkin menciptakan keunggulan kompetitif. Penyesuaian aplikasi berpaket kadang-kadang dapat mengatasi masalah dan keterbatasan ini.

5. Choose A Development Approach

Seperti yang telah disebutkan dalam gambaran umum, ada tiga pendekatan dalam pengembangan DSS yang dibahas dalam literatur IS dan DSS serta digunakan oleh praktisi. Pendekatan atau metodologi ini telah diberi berbagai nama. Secara mendasar, kita mulai dengan fokus pada keputusan dan proses pengambilan keputusan dalam langkah-langkah perancangan yang berorientasi pada keputusan; kemudian, seorang manajer proyek atau pengguna akhir melaksanakan metodologi pengembangan yang lebih atau kurang terstruktur.

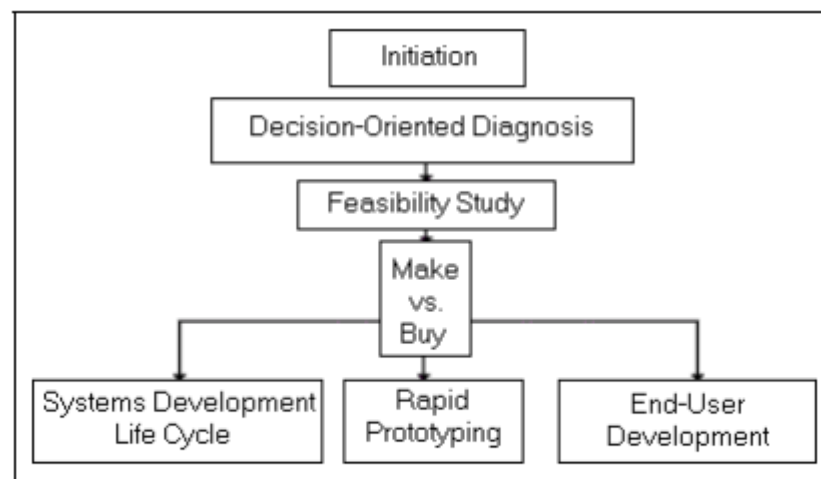


Figure 4.1. A DSS Design and Development Hierarchy.

Gambar 4.1 menunjukkan hierarki proses yang direkomendasikan untuk perancangan dan pengembangan DSS. Proses dimulai dengan diagnosis berorientasi keputusan dan analisis kelayakan, lalu berlanjut ke pengembangan DSS yang diusulkan, baik dalam perusahaan maupun melalui outsourcing, menggunakan salah satu dari tiga pendekatan pengembangan. Mari kita telaah pendekatan alternatif ini.

Pendekatan Siklus Hidup Pengembangan Sistem

Pendekatan siklus hidup pengembangan sistem (SDLC) didasarkan pada serangkaian langkah formal, termasuk tujuh langkah berikut:

Konfirmasi kebutuhan pengguna;

- Analisis sistem;
- Desain sistem;
- Pemrograman;
- Pengujian;
- Implementasi; dan
- Penggunaan dan Evaluasi.

Meskipun versi SDLC yang berbeda-beda bervariasi dalam jumlah langkah yang tepat dan definisi rinci langkah-langkah tersebut, langkah-langkah yang terdaftar di atas menggambarkan pendekatan ini. Perancangan berorientasi keputusan mulai mengatasi kebutuhan pengguna, tetapi dalam SDLC, kebutuhan pengguna harus didefinisikan secara rinci.

Pendekatan formal SDLC terkadang disebut model "waterfall" karena aliran berurutan dari satu langkah ke langkah lainnya. Setiap langkah formal diakhiri dengan penyusunan laporan kemajuan yang harus direview dan disetujui. Para reviewer termasuk calon pengguna sistem dan pengembang. Misalnya, pada Langkah 5, calon pengguna memverifikasi bahwa fungsi dan kemampuan yang didokumentasikan dan antarmuka pengguna memenuhi kebutuhan mereka. Pengembang memverifikasi bahwa antarmuka internal sistem telah didefinisikan dengan konsisten dan memenuhi semua persyaratan teknis.

Ketika pendekatan SDLC pertama kali diformalkan pada pertengahan tahun 1970-an, ia memberikan struktur dan disiplin kepada pengembang sistem. Segera, pendekatan ini banyak diadopsi untuk pengembangan TPS berukuran besar. SDLC khususnya umum ketika hubungan kontraktual formal ada antara pengembang sistem aplikasi dan pengguna akhirnya karena menyediakan bukti tertulis yang dapat digunakan untuk menyelesaikan perselisihan.

Pengembangan DSS berukuran besar yang bersifat enterprise-wide seringkali merupakan proyek yang sangat kompleks. Proses pengambilan keputusan organisasi kompleks, dan mengkomputerisasi sistem untuk mendukungnya dapat meningkatkan kompleksitas tersebut. Menggunakan metodologi seperti SDLC memberikan cara bagi

organisasi bisnis untuk mendekati pengembangan DSS berukuran enterprise secara sistematis.

Ketika pendekatan SDLC digunakan, maka rencana proyek harus dipersiapkan dengan cermat. Ketika mengembangkan kebutuhan, lebih baik dimulai dengan menentukan kebutuhan semua pengguna potensial. Kemudian, analis harus mengidentifikasi output yang akan memenuhi kebutuhan tersebut. Persyaratan teknis harus mengikuti persyaratan logis, dan kendala harus diidentifikasi untuk semua komponen sistem DSS. Persyaratan ini harus didokumentasikan dengan cermat dan direview oleh pengguna yang dituju.

Beberapa alternatif mungkin ada untuk memenuhi kebutuhan yang diidentifikasi selama langkah-langkah kebutuhan dan desain. Setiap alternatif ini harus direview dengan cermat dan yang terbaik dipilih. Pilihan lain yang harus dibuat adalah keputusan "membuat atau membeli". Jika pengembangan in-house tidak dipilih, mungkin diperlukan permintaan penawaran (RFP). Selama tahap desain, proses teknis harus dikelola, orang dan prosedur harus disiapkan, dan rencana pemasangan harus dikembangkan.

Dalam banyak situasi, pendekatan SDLC berkepanjangan terlalu kaku untuk membangun DSS, terutama DSS yang persyaratannya berubah dengan cepat. Persyaratan pengguna, yang disepakati pada tahap pertama proses, dijelaskan secara ketat dengan SDLC. Perubahan signifikan apa pun akan memulai kembali seluruh siklus pengembangan, karena dokumen persyaratan selanjutnya didasarkan pada kebutuhan pengguna yang disepakati. Perubahan seringkali mahal; sebenarnya, SDLC membatasi perubahan dalam DSS daripada mengakomodasinya.

Prototyping Cepat

Semua versi berbeda dari prototyping cepat dapat mengakomodasi dan bahkan mendorong perubahan dalam persyaratan DSS yang diusulkan. Metodologi prototyping biasa biasanya mencakup lima langkah:

- Identifikasi kebutuhan pengguna.
- Mengembangkan dan menguji prototipe DSS iterasi pertama.
- Membuat prototipe DSS iterasi berikutnya.
- Menguji prototipe DSS dan kembali ke langkah 3 jika diperlukan.
- Uji coba piloting, implementasi bertahap, atau penuh.

Pendekatan pengembangan prototyping berkembang sebagai respons terhadap kekurangan dan batasan pendekatan SDLC yang dirasakan. Dalam pendekatan pengembangan prototyping, analis DSS duduk bersama pengguna potensial dan mengembangkan persyaratan.

6. DSS Project Management

Pergeseran dari eksplorasi informal tentang saran atau keinginan untuk DSS menjadi proyek formal adalah langkah penting. Seorang sponsor eksekutif harus mendorong penunjukan seorang manajer proyek ke proyek tersebut. Tugas awal dari manajer proyek termasuk diagnosis, studi kelayakan, dan definisi tujuan serta ruang lingkup proyek yang diusulkan. Setelah langkah-langkah ini selesai, sponsor eksekutif perlu memutuskan untuk melanjutkan proyek atau menunda pekerjaan lebih lanjut pada proyek tersebut. Tergantung pada ruang lingkup proyek DSS, seorang sponsor eksekutif mungkin dapat mendanai proyek secara langsung, atau pendanaan dapat dianggarkan sebagai bagian dari perencanaan bisnis dan sistem informasi. Semakin besar ruang lingkup proyek yang diusulkan, semakin penting untuk memperoleh persetujuan dan dukungan yang luas terhadapnya.

Tujuan dari proyek DSS berukuran besar harus didorong secara strategis, harus memiliki dukungan eksekutif yang kuat, dan harus memenuhi kebutuhan bisnis. Proyek berukuran besar dapat mendapatkan manfaat dari adanya co-managers: seorang manajer bisnis dan seorang manajer teknis. Jika co-managers ditunjuk, panduan otoritas dan tanggung jawab yang jelas harus ditetapkan.

Setelah proyek disetujui, maka metodologi dan rencana proyek perlu dikembangkan dan tim proyek harus dirangkai. Jika proyek akan dioutsourcing, maka proses perlu dikembangkan untuk membuat permintaan penawaran dan kemudian mengevaluasi proposal yang diajukan. Jika pengembangan akan dilakukan secara internal, alat pengembangan dan masalah teknis perlu diselesaikan. (Analisis kelayakan seharusnya telah menentukan apakah proyek dapat diselesaikan secara internal.)

Kebutuhan pengguna perlu ditentukan dengan cukup rinci. Untuk proyek-proyek besar, arsitektur DSS harus ditentukan dan segala perubahan atau penambahan terhadap infrastruktur Sistem Informasi dan Teknologi Informasi (SI/TI) harus direncanakan. Setelah persiapan awal penting ini selesai, maka desain sistem atau prototyping dapat dilakukan. Tugas proyek tidak akan selesai dalam urutan sederhana, linear, dan manajer proyek harus mengelola proyek secara aktif. Setiap kali

memungkinkan, manajer proyek dan, dalam beberapa kasus, seorang co-manajer proyek dari area bisnis yang paling terpengaruh harus berkonsultasi dan bekerja sama dengan pengguna potensial lainnya. Manajer proyek harus tetap menginformasikan sponsor eksekutif. Jika masalah sedang terjadi atau mungkin terjadi, sponsor perlu diingatkan.

Manajer proyek harus mengidentifikasi tugas-tugas yang harus diselesaikan, sumber daya yang dibutuhkan, dan hasil proyek. Hasil proyek sangat penting untuk memantau kemajuan proyek. Batu loncatan atau peristiwa penting dalam proyek juga sering diidentifikasi untuk membantu manajer non-teknis memantau proyek. Chief Information Officer (CIO) dari sebuah perusahaan dan satu atau lebih manajer bisnis biasanya memantau kemajuan proyek DSS berukuran besar atau berdampak tinggi. Manajer mengharapkan hasil dari proyek DSS. Memahami dan memenuhi harapan manajer yang akan menggunakan DSS adalah bagian yang paling penting dan paling sulit dari pekerjaan manajer proyek DSS.

Manajer proyek mendefinisikan rencana proyek dan mengelola aktivitas harian yang terkait dengan proyek tersebut. Dia juga mengoordinasikan sumber daya proyek, anggaran proyek, pelaporan status, perubahan dalam kebutuhan dan tugas, hubungan dengan vendor, serta hubungan dengan sponsor, skeptis, dan staf SI. Manajer proyek DSS dapat berasal dari sistem informasi atau dari departemen fungsional dan memerlukan keterampilan teknis yang kuat, keterampilan interpersonal yang luar biasa, sikap tegas, dan pengetahuan tentang bisnis.

Outsourcing

Outsourcing melibatkan kontrak dengan konsultan eksternal, perusahaan perangkat lunak, atau biro jasa untuk melakukan analisis sistem, pemrograman, atau aktivitas pengembangan DSS lainnya. Outsourcer harus dievaluasi sebagai aset jangka panjang dan sebagai sumber nilai berkelanjutan bagi perusahaan. Waktu dan sumber daya perlu diperuntukkan untuk mengelola hubungan tersebut dan memaksimalkan nilainya. Pelanggan membutuhkan manajer proyek untuk mengelola hubungan outsourcing. Niatnya harus menjaga hubungan tersebut selama masih memberikan nilai kepada pelanggan. Seiring waktu, aliansi teknologi baru mungkin diperlukan karena teknologi dan organisasi berubah.

7. DSS Project Participant

Sebuah DSS yang kompleks, yang dibangun menggunakan pendekatan SDLC atau prototyping, memerlukan pendekatan berbasis tim dalam pengembangannya. Setelah sistem dikembangkan, sebuah grup mungkin juga perlu untuk menjaga sistem tersebut. Beberapa DSS berukuran besar dibangun dengan tim dua atau tiga orang, atau dengan kelompok yang lebih besar, sepuluh orang atau lebih. Anggota tim DSS berasal dari berbagai area dalam sebuah organisasi, termasuk grup IS (Sistem Informasi).

Setiap proyek pengembangan DSS memerlukan campuran keterampilan yang saling melengkapi. Biasanya, tidak semua keterampilan yang dibutuhkan dapat ditemukan dalam satu orang. Jadi, dalam kebanyakan situasi, perlu untuk mengumpulkan campuran kontributor yang tepat untuk tim proyek DSS. Peran pengembangan DSS kunci yang diidentifikasi oleh Sprague (1980), O'Neil dkk. (1997), dan orang lainnya tercantum di bawah ini dalam urutan peningkatan keahlian teknis.

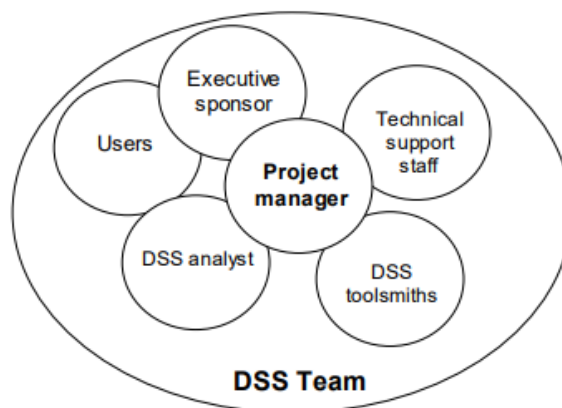


Figure 4.2. Participants on a DSS Development Team.

Gambar 4.2 merangkum berbagai peran tersebut. Seorang individu tertentu mungkin diberi lebih dari satu peran.

- **Manajer Proyek:** Ini adalah manajer bisnis atau teknis yang dapat mengorganisir dan mengelola sumber daya yang diperlukan untuk menyelesaikan proyek DSS.
- **Pendukung Eksekutif atau Pemimpin Proyek:** Ini adalah manajer senior yang memiliki akses ke eksekutif senior lainnya dan memiliki pengaruh untuk membantu menyelesaikan masalah sumber daya dan politik utama. Sponsor terkadang secara aktif terlibat dalam tugas pengembangan.
- **Potensi Pengguna DSS:** Ini adalah orang yang mengambil keputusan yang akan didukung oleh DSS yang diusulkan. Pengguna seringkali adalah orang-orang non-teknis di area fungsional bisnis seperti pemasaran dan keuangan.

- **Analisis DSS:** Ini adalah spesialis MIS yang bertindak sebagai perantara atau penghubung antara pengguna dan pengembang DSS. Seorang analisis DSS dapat membuat keputusan tentang alat perangkat lunak yang akan digunakan, platform perangkat keras yang akan digunakan, model dan/atau basis data yang akan digunakan dalam DSS, dan bagaimana mereka akan diintegrasikan satu sama lain. Biasanya ini adalah orang yang berpengalaman yang memahami masalah bisnis dan teknologi yang tersedia. Seorang analisis DSS sering bekerja mengumpulkan persyaratan, menganalisis solusi, menulis spesifikasi, menjaga informasi produk, serta membantu dalam pelatihan dan dukungan dokumentasi. Seorang analisis DSS sering bekerja dengan pengguna untuk menentukan dan mendokumentasikan persyaratan sistem untuk Sistem Pendukung Keputusan. Seorang analisis DSS juga dapat membantu dalam mendesain ulang proses bisnis untuk lebih memanfaatkan Sistem Pendukung Keputusan yang berbasis komputer. Orang ini terkadang juga mengambil peran sebagai manajer proyek.
- **Staf Dukungan Teknis:** Sejumlah profesional MIS terlibat sebagai staf dukungan teknis, termasuk arsitek gudang data, spesialis jaringan, arsitek aplikasi, peneliti operasi, dan pengembang. Seorang pemodel data dan analisis kualitas data sering terlibat dalam membangun DSS yang berfokus pada data. Analisis kualitas data peduli dengan integrasi data, metadata, dan pemurnian data. Seorang administrator basis data adalah bagian integral dari tim pengembangan untuk proyek DSS berbasis data. Administrator data, administrator sistem, dan spesialis jaringan sering dikonsultasikan dalam proyek DSS dan mungkin bergabung dengan tim pengembangan untuk beberapa proyek.
- **Pengembang Alat DSS:** Ini adalah spesialis dengan alat dan teknologi yang akan digunakan dalam pembangunan DSS dan paket yang akan digabungkan untuk membuat DSS. Dia adalah ahli dalam alat dan paket ini, dan penggunaan mereka yang efektif. Ini adalah orang yang menciptakan kemampuan dasar dan mengintegrasikan paket yang ada menjadi satu sistem keseluruhan serta melakukan pemrograman kustom yang berkontribusi langsung pada fungsionalitas DSS. Tanggung jawabnya dimulai dari paket-paket yang akan menjadi bagian dari DSS dan berakhir dengan penyelesaian DSS tertentu.

Komposisi tim DSS dapat berubah selama siklus pengembangan, jadi manajer proyek dan pembangun DSS perlu memberikan arahan dan motivasi untuk tim DSS.

Selain itu, sponsor eksekutif perlu mempertahankan komitmen aktif terhadap proyek. Kehilangan sponsor proyek dapat merugikan dan bahkan menghancurkan proyek DSS.

8. Conclusion

Pada tahun 1985, Jack Hogue dan Hugh Watson melakukan survei terhadap manajer-manajer di organisasi-organisasi yang menggunakan DSS. Masing-masing peserta adalah pengguna DSS aktif. Dua pertiga dari organisasi-organisasi tersebut telah membangun DSS mereka dengan pendekatan evolusioner, prototyping, dan organisasi-organisasi lainnya telah menggunakan pendekatan SDLC yang lebih formal. Tampaknya, jika DSS mendukung manajer-manajer di seluruh perusahaan atau memerlukan data yang berskala perusahaan, maka pendekatan SDLC lebih mungkin digunakan. Pendekatan evolusioner digunakan untuk sistem yang berukuran lebih kecil di mana alat pengembangan DSS tersedia. Sembilan dari delapan belas perusahaan menggunakan generator DSS untuk mengembangkan sistem mereka. Temuan ini kemungkinan merupakan deskripsi praktik saat ini.

Ketika manajer dapat menentukan persyaratan informasi sebelumnya, maka pendekatan SDLC lebih mungkin digunakan. Hogue dan Watson juga menemukan bahwa ketika spesialis SI mengembangkan DSS, maka langkah-langkah SDLC lebih mungkin diikuti. Manajer senior melaporkan bahwa mereka paling terlibat dalam ide, persyaratan informasi, dan langkah penerimaan yang terkait dengan membangun DSS. Manajer menengah melaporkan bahwa mereka agak terlibat dalam semua langkah yang terlibat dalam membangun DSS yang mereka gunakan. Ketika prototyping dan desain evolusioner digunakan, manajer melaporkan lebih banyak keterlibatan dalam proses desain dan pengembangan. Grup SI biasanya terlibat dalam membangun DSS, tetapi staf dari departemen SI jarang berada dalam peran kepemimpinan. Pengguna potensial DSS biasanya mengambil peran kepemimpinan.

Pendekatan desain dan pengembangan DSS yang digunakan untuk proyek DSS baru harus bergantung pada jumlah data yang dibutuhkan dan sumbernya, jumlah pengguna yang direncanakan, model dan alat analisis yang digunakan, dan jumlah penggunaan yang diantisipasi. Banyak DSS kecil yang bersifat khusus dapat dibangun dengan cepat menggunakan pengembangan oleh pengguna akhir atau prototyping cepat. DSS berukuran besar yang berfokus pada perusahaan dibangun dengan menggunakan alat yang canggih serta pendekatan analisis dan pengembangan sistem yang sistematis dan terstruktur. Membuat lingkungan DSS perusahaan secara keseluruhan tetap merupakan tugas yang kompleks dan evolusioner. DSS perusahaan

yang berskala besar secara inevitabel menjadi bagian besar dari infrastruktur sistem informasi keseluruhan perusahaan. Meskipun terdapat perbedaan pengembangan yang signifikan yang diakibatkan oleh cakupan dan tujuan DSS, semua DSS memiliki komponen teknis yang serupa dan memiliki tujuan yang sama, yaitu mendukung pengambilan keputusan.

Sejumlah penulis menyarankan bahwa persepsi tentang kegunaan dan kemudahan penggunaan dari sebuah sistem informasi atau DSS adalah penentu utama penggunaannya. Manajer SI dapat memengaruhi baik persepsi tentang kegunaan maupun persepsi tentang kemudahan penggunaan sistem baru dengan menggunakan proses pengembangan yang partisipatif. Staf SI perlu membangun "pertukaran sosial" yang berarti dengan pengguna potensial, dan pengembang DSS harus responsif terhadap permintaan, pertanyaan, dan kebutuhan pengguna.

Lebih banyak penelitian diperlukan tentang efektivitas pendekatan desain dan pengembangan DSS. Namun, secara umum, profesional SI harus menggunakan proses desain yang berorientasi pada keputusan dan kemudian menggunakan proses prototyping cepat atau SDLC. DSS yang dikembangkan oleh pengguna akhir dapat memuaskan dan ekonomis, dan staf SI harus bekerja sama untuk mendukung pengembangan semacam itu daripada menghalanginya ketika terlihat sesuai. Prototyping cepat berguna dalam membangun banyak jenis DSS, tetapi SDLC memiliki peran dalam pengembangan DSS yang kompleks, berjangkauan, berbasis data, skala perusahaan. Analis DSS dan manajer perlu mengenal semua pendekatan untuk membangun DSS. Ada beberapa generalisasi yang dapat diambil tentang Desain dan Pengembangan Sistem Pendukung Keputusan (DSS):

- Ketika sebuah ide proyek diajukan, fokus harus pada deskripsi dan diagnosis pengambilan keputusan serta analisis keputusan dan proses yang terlibat. Pendekatan ini disebut sebagai diagnosis berorientasi pada keputusan.
- Setelah diagnosis, langkah berikutnya adalah melakukan studi kelayakan, dan dalam banyak situasi, menyiapkan laporan kelayakan.
- Jika proyek tampak layak, maka manajer dan staf Sistem Informasi (SI) perlu memutuskan apakah akan membangun atau membeli DSS yang diusulkan. Dalam banyak situasi, solusi akan disesuaikan dengan kebutuhan DSS tersebut.
- Secara umum, DSS yang berbasis model dan berbasis pengetahuan dibangun dengan menggunakan prototyping cepat. DSS yang berbasis data dan dokumen

dibangun menggunakan prototyping cepat atau pendekatan Siklus Hidup Pengembangan Sistem (SDLC) tergantung pada kompleksitas dan skala sistem tersebut. DSS yang berbasis komunikasi biasanya dibeli dan diinstal di komputer perusahaan.

- Akhirnya, manajer perlu mengembangkan pemahaman yang komprehensif tentang cara merancang dan mengembangkan berbagai jenis DSS. Pada akhirnya, manajer senior bertanggung jawab untuk memastikan bahwa proyek DSS mendukung tujuan bisnis dan memberikan manfaat bagi organisasi. Kesesuaian proses desain dan pengembangan dengan kegiatan pelatihan pengguna dalam pengoperasian DSS baru akan menentukan apakah tujuan bisnis akan tercapai dan manfaat akan dirasakan.