

PENUNJANG SISTEM (Support Systems)

1. Definisi

Penunjang Sistem merupakan penyempurnaan sistem secara terus menerus setelah sistem tersebut digunakan. Aktivitas penyempurnaan ini termasuk penyempurnaan program dan sistem. Penunjang sistem seringkali memerlukan pengembang dalam melihat kembali aktivitas yang dijalankan sewaktu analisis, desain dan implementasi sistem.

2. Penunjang Sistem : Penyimpanan data

Terdapat 3 jenis penyimpanan data bagi suatu sistem, yaitu:

- Repository (simpanan terpusat): Tempat menyimpan semua model sistem dan spesifikasi sistem yang terperinci.
- Program Library (perpustakaan pemrograman): Tempat menyimpan pemrograman aplikasi yang benar-benar sedang digunakan.
- Database (Pangkalan data): Tempat menyimpan data operasional yang dibangun dan disempurnakan oleh pemrograman aplikasi yang dihasilkan.

3. Aktivitas Penunjang sistem

Penunjang sistem biasanya dilakukan oleh pendesain dan pengembang sistem dan dibantu oleh pengguna sistem. Walaupun sebuah sistem atau aplikasi sistem telah dikembangkan, didesain dan diuji dengan sempurna, namun kesalahan atau perbaikan tidak dapat dielakkan. Usaha membetulkan keadaan ini dipanggil penyempurnaan sistem atau pemrograman.

Tujuan dari penyempurnaan sistem, adalah:

- Membuat perubahan terhadap pemrograman yang ada dimana terdapat perbaikan sewaktu desain dan implementasi sistem.

- Aktivitas menentukan dan menyetujui masalah tersebut dijalankan oleh analis sistem dan pemrogram yang melibatkan pengguna.
- Pemrograman yang bermasalah akan diperoleh dari penyimpanan data librari dan kemudian diperbaiki. Ada kemungkinan projek bisa ditunda sementara.

3.1 Membetulkan Kesalahan

(a). Memahami aplikasi dan pemrograman

Biasanya penyempurnaan sistem ini tidak dilakukan oleh individu yang sama yang menulis pemrograman. Oleh itu individu tersebut perlu memahami keseluruhan pemrograman dan bagian-bagian yang bermasalah. Aktiviti ini akan memerlukan waktu yang banyak apabila; struktur modular yang lemah, logika pemrograman yang tidak terstruktur, kode pemrograman yang tergantung dan dokumentasi yang tidak memuaskan.

Aktivitas memahami aplikasi dan pemrograman bertujuan:

- untuk menentukan bagaimana pemrograman beroperasi dan bagian-bagian mana yang tidak beroperasi,
- memahami variabel pemrograman,
- menentukan di mana dan bagaimana variabel berfungsi,
- menentukan implikasi perubahan yang bakal terjadi terhadap semua pemrograman dan
- memahami pemrograman juga akan menentukan waktu dan tenaga yang diperlukan untuk membetulkan kesalahan tersebut.

(b). Mengedit dan menguji pemrograman

Terdapat perbedaan diantara mengedit pemrograman baru dan mengedit pemrograman yang sudah ada. Pemrograman dibagi kepada bagian pemrograman yang *stand alone* dan pengujian dijalankan bagi setiap bagian.

Pengujian terhadap pemrograman adalah sebagai berikut:

- Ujian Sistem : menentukan keseluruhan aplikasi termasuk yang telah diperbaiki fungsinya.
- Ujian Regression : melihat implikasi perubahan yang dilakukan ke atas pemrograman dan waktu tindak balas sebelum dan setelah perubahan dilakukan dengan menggunakan data ujian dan prestasi dari waktu ke waktu.

Kontrol versi adalah proses dimana librari (selalunya berdasarkan software) mengontrol perubahan yang dilakukan ke dalam pemrograman.

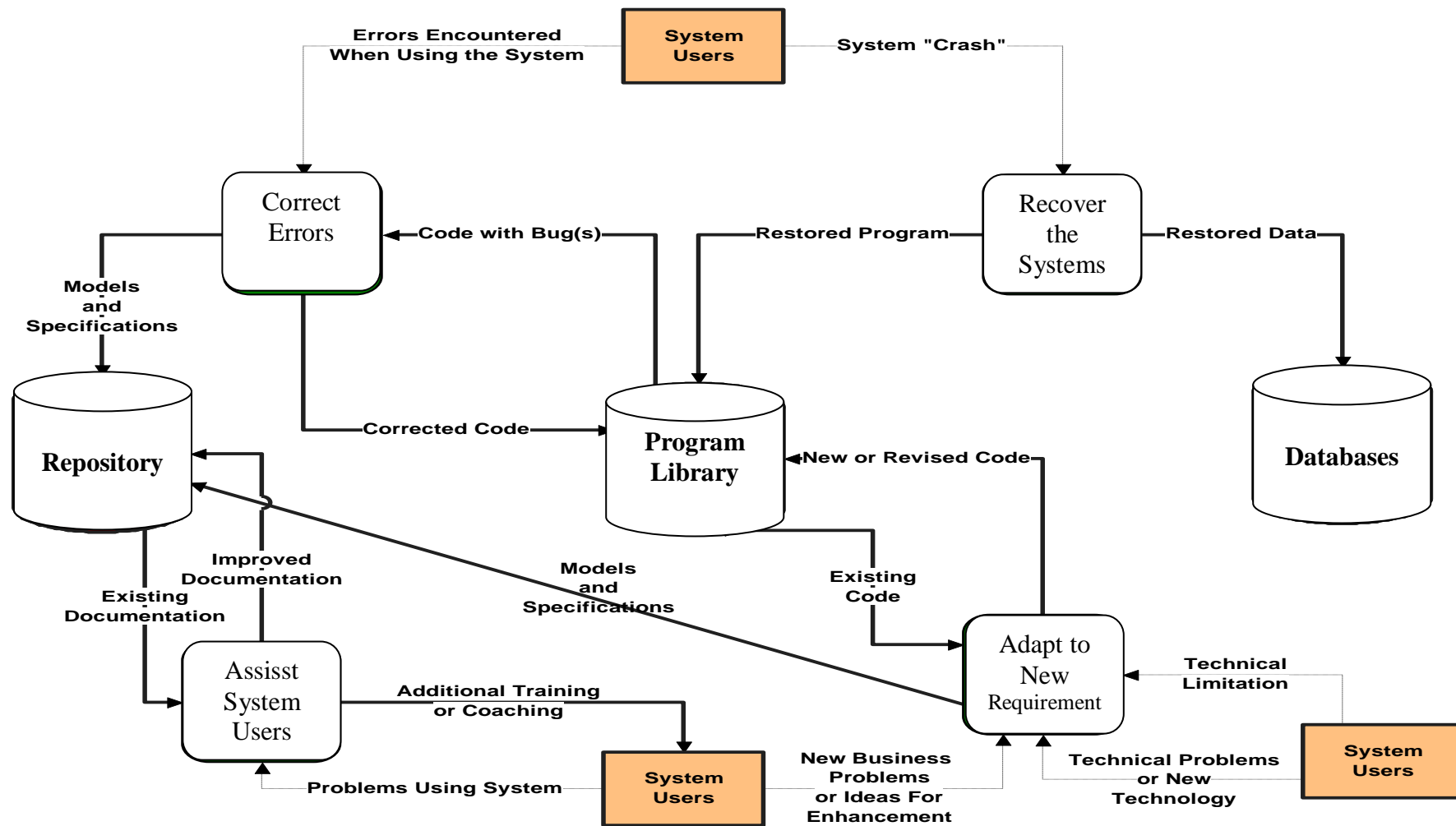
(c). Mengemaskini Dokumentasi

Biaya yang tinggi dalam penyempurnaan sistem adalah disebabkan oleh kegagalan menyempurnakan aplikasi dan dokumentasi. Jika dokumentasi aplikasi telah berubah, librari pemrograman perlu diubah juga. Dan perubahan informasi terbaru di dalam repositori dan librari pemrograman membantu programmer dan analisis sistem mengurangi waktu yang diambil untuk memahami aplikasi sewaktu penyempurnaan sistem.

3.2 Perbaikan Sistem: Mengatasi Kegagalan Sistem

Kegagalan sistem adalah diluar perkiraan semua pihak, mungkin diakibatkan oleh terjadinya pemrograman hang, abort atau crash. Kejadian ini bisa mengakibatkan data hilang. Apabila hal ini terjadi, maka yang harus dilakukan adalah:

- Analisis sistem berusaha untuk memulihkan sistem sendiri
- Menghubungi operasional sistem untuk membetulkan sistem
- Menghubungi pengelola data untuk memperbaiki fail data atau pangkalan data yang hilang atau rosak
- Menghubungi pengelola rangkaian untuk memperbaiki masalah rangkaian apakah local, wide atau internet
- Menghubungi tenaga teknisi atau pelayan vendor untuk memperbaiki masalah hardware.



Gambar 1 : Types of System Support

3.3 Perubahan Sistem

Sistem perlu diubah berdasarkan kepada unpan balik masalah baru bisnes, informasi spesifikasi baru atau saran baru untuk kemajuan sistem. Librari aplikasi perlu dilihat untuk menentukan aplikasi dan pemrograman yang memerlukan biaya yang tinggi atau yang paling sukar untuk diperbaiki. Sistem perbaikan ini dalam istilah sistem disebut rekayasa lagi (recover the system).

Tujuan dari perbaikan sistem adalah

- sistem agar sesuai dengan teknologi terbaru.
- membetulkan sistem sebelum sistem tersebut gagal berfungsi
- membuatkan sistem mudah dibetulkan apabila sistem tersebut gagal beroperasi

(a) Meneliti saran perbaikan

Menentukan apakah ada aktivitas sebagai berikut perlu dilakukan: masalah bisnes baru, idea untuk perbaikan dan hambatan teknikal. Berdasarkan analisa model sistem pada saat ini, aktivitas yang harus dilakukan adalah:

- Menentukan keperluan bisnes baru, modal atau biaya dikembalikan kepada analisa sistem.
- Menentukan kebutuhan teknikal baru dan kembali ke desain sistem
- Menentukan kebutuhan pemrograman baru
 - Menulis pemrograman yang baru dan mudah
 - Menstruktur lagi pangkalan data atau fail
 - Menganalisis librari pemrograman dan biaya penyempurnaan

Di dalam menganalisis librari pemrograman dan biaya penyempurnaan bisa menggunakan matriks sistem yang merupakan tolak ukur kualitas dan produktivitas sistem. Contoh matriks sistem, adalah:

- Ikatan aliran kontrol : jumlah aliran logika yang bersilang. Sebaiknya pemrograman tidak mempunyai ikatan aliran kontrol.
- Siklus kompleks : jumlah aliran yang unik. Sebaiknya aliran ini lebih kecil.

(b) Systems Enhancement and Reengineering

Terdapat tiga jenis perbaikan pemrograman, yaitu:

- Organisasi lagi kode atau simbol pemrograman : penstrukturan lagi logika pemrograman
- Mengubah kode atau simbol pemrograman : Tukar kode dan simbol ke dalam bahasa pemrograman yang lain
- Meringkas kode dan simbol pemrograman : meringkas bagian-bagian kode dan simbol untuk mengembangkan pemrograman ke dalam sub-pemrograman yang berlainan.