

Antarmuka Grafis (GUI) pada KDE sebagai Aplikasi Desktop Environment

Asfihani (asfik@cakraweb.com)

17 Januari 2004

Abstract

KDE (K Desktop Environment) merupakan salah satu *windows manager* di sistem operasi berbasis UNIX. Pembuatan aplikasi yang berjalan di KDE harus mengikuti standar antarmuka yang ramah dan efisien bagi penggunaanya. Tujuan dari penulisan dokumen ini adalah untuk memberikan gambaran mengenai sistem antarmuka (*interface*) aplikasi-aplikasi yang terdapat di KDE.

Contents

1	Pendahuluan	2
1.1	Linux	2
1.2	KDE	3
2	Pengantar antarmuka grafis (GUI) pada KDE	4
2.1	Basic	5
2.1.1	GUI Usage	5
2.1.2	Terminology	6
2.1.3	Windows	7
2.1.4	Labels	11
2.1.5	Settings	12
2.1.6	Systray	13
2.1.7	Interface example	14
2.2	Menu	15
2.2.1	File	16
2.2.2	Edit	18
2.2.3	View	18
2.2.4	Go	19
2.2.5	Bookmarks	20
2.2.6	Tools	20
2.2.7	Settings	20

2.2.8	Help	21
2.3	Toolbar	21
2.4	Statusbar	22
2.5	Content Area	23
2.6	Dialogs	23
2.6.1	Simple Dialogs	24
2.6.2	Complex Dialogs	26
2.6.3	Standard Dialogs	27
2.7	Keys	27
2.7.1	Accelerators	27
2.7.2	Shortcuts	28
2.7.3	Shift Modifier	29
2.7.4	Cursor	29
2.7.5	Selection	30
2.7.6	Dialog Navigation	30
2.7.7	Completion	30
2.8	Mouse	31
2.9	Drag and Drop	31
3	Daftar Pustaka	32

1 Pendahuluan

1.1 Linux

Pada saat ini, penggunaan *software* (perangkat lunak) sistem operasi pada komputer personal yang bersifat komersial memang masih mendominasi hampir 90% pengguna komputer di rumah. Namun, software sistem operasi lain yang menggunakan lisensi GPL (<http://www.gnu.org/copyleft/gpl.html>) mulai naik dan menunjukkan peningkatan yang signifikan. Lihatlah sistem operasi Linux yang dikenalkan oleh Linus Torvalds, seorang mahasiswa di Universitas Helsinki, Finlandia pada tahun 1991.

Pada perkembangannya, Linux yang menjadi perangkat lunak yang mulai digunakan sebagai *desktop*. Pada tahap awal, dukungan terhadap *desktop environment* (lingkungan desktop) yang ramah terhadap pengguna memang masih sedikit. Namun pada saat ini, jumlah *desktop environment* atau yang sering disebut dengan *windows manager* di Linux sudah semakin banyak dan menarik. Mulai dari yang menawarkan fitur lengkap seperti KDE (<http://www.kde.org>) dan GNOME (<http://www.gnome.org>), sampai dengan fitur sederhana tapi lebih ringan, ringkas dan sederhana seperti XFCE (<http://www.xfce.org>), icewm (<http://www.icwm.org>), dan lain- lainnya.

1.2 KDE

Sejarah KDE

Pada bulan Oktober 1996, Matthias Ettrich, seorang mahasiswa dari Universitas Eberhard-Karls di Tuebingen Jerman, mengirimkan *posting* artikel di *newsgroup* yang berisi undangan untuk bergabung dengan proyek *desktop environment* yang diberi nama KDE (K Desktop Environment)¹. Matthias Ettrich merasa bahwa perlu adanya *desktop environment* untuk di sistem operasi berbasis UNIX bagi pengguna *end user*. Dalam postingnya, Matthias Ettrich juga menjelaskan penggunaan *library QT* dari Trolltech (<http://www.trolltech.com>).

Tentang KDE

KDE adalah sebuah aplikasi *desktop environment* untuk sistem operasi berbasis UNIX (termasuk Linux, FreeBSD, NetBSD dan lain-lainnya). Fungsinya adalah untuk mengurangi penggunaan *command line* yang mungkin terlalu rumit bagi pengguna yang awam di sistem operasi berbasis UNIX tersebut, sehingga pengoperasiannya semudah mengoperasikan *desktop* pada lingkungan Windows maupun MacOS.

Pada saat tulisan ini dibuat versi KDE yang paling baru adalah KDE 3.1.4 (rilis), sedangkan untuk versi pengembangan adalah KDE 3.2 Beta 2². Pada saat ini, hampir semua distribusi besar Linux (misalnya Redhat, SuSe dan Debian) menyertakan KDE sebagai *desktop environment*-nya.

Aplikasi KDE

Aplikasi-aplikasi yang disertakan dalam paket distribusi KDE adalah ³:

- **Konqueror Browser**

Konqueror (<http://konqueror.kde.org>) merupakan aplikasi yang terintegrasi mencakup *web browser*, *file manager* dan *document viewer*. Konqueror sudah mendukung teknologi terbaru di internet, termasuk didalamnya JavaScript, Java, HTML 4.0, CSS-1 dan -2 (Cascading Style Sheets), SSL (Secure Socket Layer for secure communications) dan Netscape Communicator plug-ins (untuk memainkan Flash, RealAudio, RealVideo dan teknologi yang serupa).

- **KOffice Office Suite**

KDE juga menyertakan aplikasi perkantoran KOffice (<http://www.koffice.org>). KOffice merupakan aplikasi suit yang berisi aplikasi *spreadsheet* (KSpread), aplikasi pengolahan gambar berbasis vektor (Karbon), aplikasi pengolah

¹<http://www.kde.org/announcements/announcement.php>

²<http://www.kde.org/download>

³<http://www.kde.org/info>

kata (KWord), aplikasi presentasi (KPresenter), serta aplikasi pengolahan diagram dan *chart* (KChart).

- **aRts Multimedia Architecture**

Aplikasi aRts (Analog Realtime Synthesizer) memungkinkan pengguna KDE untuk memainkan file multimedia dari berbagai *resource* (sumber), seperti misalnya sumber dari *network (streaming)*, maupun dari sumber yang lain seperti file lokal.

- **KIO Network Transparency**

Aplikasi KIO's network transparency memungkinkan pengguna KDE mengakses file dari berbagai macam *resource* yang ada, misalnya NFS, SMB pada Windows, halaman HTTP, direktori FTP maupun direktori LDAP. Sedangkan untuk *plugin* yang modular bisa mendukung protokol tambahan seperti IPX dan WebDav sehingga memungkinkan aplikasi lain di KDE untuk menggunakan file tersebut.

- **Customizability**

KDE menggunakan *widget* yang bisa dimodifikasi agar sesuai keinginan. KDE menyertakan lebih dari 14 model *style* (bentuk) yang memungkinkan pengguna untuk merubah tampilan *icon*, *themes*, *system sound*, *keybindings*, pilihan lebih dari 50 bahasa, dan lain-lainnya.

Sebagai catatan, jumlah baris pada *source code* yang tersimpan CVS (Concurrent Versions System) adalah sekitar 4 juta baris (bandingkan dengan jumlah baris pada *source code* kernel Linux 2.5.7 yang ”hanya” 3.7 juta baris). Jumlah *volunteer* (sukarelawan) pengembang di KDE sekitar 800 orang yang berasal dari berbagai negara di dunia ⁴.

2 Pengantar antarmuka grafis (GUI) pada KDE

Pada bab ini akan mencoba membahas bagaimana *look-and-feel* aplikasi-aplikasi di KDE. Bahasan-bahasannya adalah pada petunjuk penggunaan kontrol pada aplikasi secara umum yang biasanya terdiri dari sebuah *menubar* (menu), *toolbar* (peralatan), *working area* (tempat kerja) dan *statusbar* (status).

Meskipun demikian, tidak semua aplikasi harus memenuhi kontrol-kontrol tersebut diatas, hanya bila diperlukan saja. Penggunaan kontrol yang konsisten akan mempermudah bagi pengguna jika banyak menggunakan aplikasi yang berbeda-beda. Contoh : KNotes merupakan aplikasi KDE yang hanya terdiri dari *working area*. Desainnya merupakan tiruan dari produk Post-It notes dari 3M. Walaupun tidak memiliki *menubar* ataupun *toolbar*, namun secara umum

⁴<http://www.kde.org/whatiskde/project.php#factsandfigures>

interface-nya (*dialogs*, penggunaan *mouse*) masih mengikuti standar sebuah aplikasi di KDE.



Gambar 1. Aplikasi sederhana (KNotes).

2.1 Basic

2.1.1 GUI Usage

Kebanyakan aplikasi-aplikasi modern mempunyai *layout* (tampilan) yang hampir sama pada *window* (jendela) utamanya. Aplikasi-aplikasi tersebut biasanya menawarkan sebuah jendela dokumen dan beberapa kontrol (misalnya tombol) untuk memanipulasinya. Sedangkan kontrol untuk *action* biasanya diakses melalui *menu* dan *shortcut* yang bisa diakses melalui *toolbar* ataupun keyboard. Sedangkan hasil dari *action* terhadap dokumen tersebut akan ditampilkan dalam *statusbar*.

Beberapa hal harus dipenuhi sebelum suatu perangkat lunak bisa dikatakan ramah pengguna (*user friendly*). Sebagai perbandingan, 80%-90% perangkat lunak yang ada sekarang ini belum sepenuhnya memenuhi hal tersebut.

Agar suatu perangkat lunak bisa ramah pengguna, maka harus:

- **Sesuai keperluan**
Jangan menawarkan aplikasi yang akan membingungkan pengguna dengan fungsi-fungsi yang tidak sesuai dengan kebutuhan.
- **Bisa dipahami**
Ketika pengguna menggunakan aplikasi untuk pertama kali, maka pengguna harus mengetahui kegunaan aplikasi tersebut dan bagaimana cara menggunakannya.
- **Mudah dikendalikan**
Pengguna harus dapat mengetahui dimana dia berada.

- **Sesuai dengan harapan**
Aplikasi harus konsisten dan sesuai dengan harapan pengguna.
- **Toleran terhadap kesalahan**
Pengguna juga merupakan manusia yang bisa membuat kesalahan. Jika dimungkinkan, aplikasi harus bisa melakukan *Undo* (pengulangan ke kondisi keadaan yang diinginkan).
- **Kaya akan umpan-balik**
Aplikasi harus bisa membimbing pengguna dengan umpan-balik tentang tindakan apa yang harus dilakukan.

2.1.2 Terminology

Sebuah antarmuka pengguna (*user interface*) harus didesain berdasarkan pandangan dari sisi pengguna. Meskipun demikian, terkadang pandangan dari pembuat aplikasi (*developer*) bisa saja berbeda dari pengguna. Untuk menghindari perbedaan pandangan yang mungkin terjadi, beberapa kesepakatan harus dibuat secara bersama-sama oleh pengguna dan pembuat aplikasi.

Sebagai contoh adalah pengertian **Quit** dari sisi pengguna dan pembuat aplikasi. Pengertian “aplikasi” sendiri sering membingungkan. Dari pandangan pengguna, aplikasi adalah semua item entri yang terdapat pada *taskbar* adalah merupakan aplikasi yang sedang dijalankan. Sedangkan bagi pembuat aplikasi, setiap proses pada sistem UNIX merupakan aplikasi yang sedang berjalan. Biasanya pengguna tidak tertarik dengan proses-proses di UNIX, maka pembuat aplikasi tidak perlu memberitahukannya secara detail. Antarmuka pengguna-lah yang akan membuatnya transparan terhadap pengguna. Adalah sesuatu yang buruk dari pandangan antarmuka pengguna, jika *option* (pilihan) **Quit** merupakan tindakan untuk menghentikan semua aplikasi di UNIX. Yang benar adalah hanya menghentikan aplikasi yang dipilih oleh pengguna tersebut sedangkan aplikasi-aplikasi yang lain harus dibiarkan walaupun menggunakan proses di UNIX yang di-*share* maupun tidak.

Beberapa kesepakatan dibawah ini sangat penting agar tidak terjadi kesalah pahaman antara pengguna dan pembuat aplikasi.

Proses-proses di UNIX

- Sebuah proses di UNIX menggunakan *resources*.
- Proses-proses di UNIX bisa dilihat dengan menggunakan perintah "ps", "top" dan sejenisnya.
- Sebuah proses di UNIX bisa mengalami *crash*.

Aplikasi

- Setiap *window* yang memiliki *toolbar* dan/atau *menubar* merupakan aplikasi dari pandangan pengguna.
- Setiap aplikasi (yang sedang dijalankan) akan nampak di *taskbar*.
- Sebuah aplikasi ditampilkan dalam bentuk *icon*, dimana jika *icon* tersebut di klik maka akan memulai aplikasi tersebut. Jika sudah ada aplikasi yang sama sudah berjalan, maka tidak akan berpengaruh (tidak dijalankan).

Dokumen

- Sebuah dokumen mengandung sebuah informasi.
- Sebuah dokumen mempunyai lokasi penyimpanan bisa dalam bentuk file-name atau URL.
- Bisa dibuka oleh suatu aplikasi.
- Hanya dokumen yang sedang dibuka yang bisa dimodifikasi.
- Perubahan bisa disimpan.
- Suatu dokumen yang sedang dibuka bisa ditutup.
- Dokumen ditampilkan dalam bentuk *icon*, jika *icon* tersebut diklik maka akan membuka aplikasi yang berkaitan.

Window

- Sebuah *window* adalah bagian dari aplikasi yang menampilkan dokumen, informasi, dan/atau *tools* (alat-alat) kepada penggunanya.

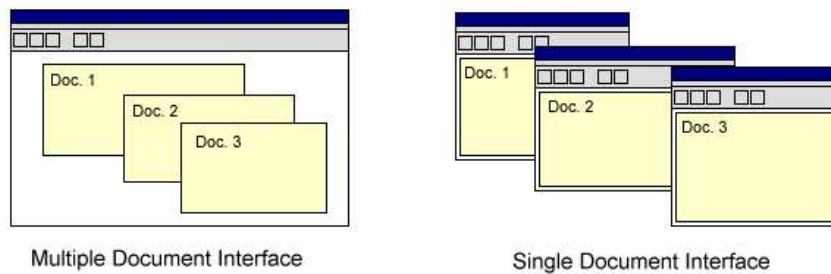
2.1.3 Windows

KDE, seperti halnya sistem yang lain yang menggunakan *window*, menampilkan aplikasinya dalam sebuah *window* yang *moveable* (bisa dipindah-pindahkan) dan *resizeable* (bisa disesuaikan ukurannya), dan bisa ditempatkan pada lokasi yang sesuai dengan keinginan penggunanya.

KDE tidak menggunakan MDI (Multi Document Interface). MDI dikenalkan beberapa tahun yang lalu diadopsi secara luas oleh Microsoft Corporation pada produk-produk Windows mereka. MDI menggunakan sebuah *window* induk (utama) yang berisi seluruh aplikasi-aplikasi kecil didalamnya. Dalam hal ini,

aplikasi-aplikasi kecil tersebut adalah *document windows*, *toolbar windows*, *help windows* dan seterusnya. Segala kejadian yang terjadi dilayar akan ditampilkan pada *window* utamanya. Contoh aplikasi yang menggunakan MDI adalah aplikasi File Manager pada sistem operasi Windows 3.1. Selama beberapa tahun MDI cukup memudahkan bagi para pembuat aplikasi. Namun dalam perkembangannya, pengguna menginginkan kebebasan dalam mengatur *window* aplikasi mereka, seperti misalnya mengubah ukuran ataupun meletakkannya sesuai dengan keinginan.

KDE menggunakan SDI (Single Document Interface). Sebuah aplikasi berbasis SDI tidak memiliki *window* induk (utama) yang berisi semua *window* aplikasi didalamnya. *Window* aplikasi merupakan bagian dari *X root window*, yang berarti mereka dikendalikan oleh *window manager* yang bisa dipindahkan dalam *desktop environment* sesuai dengan keinginan penggunanya.

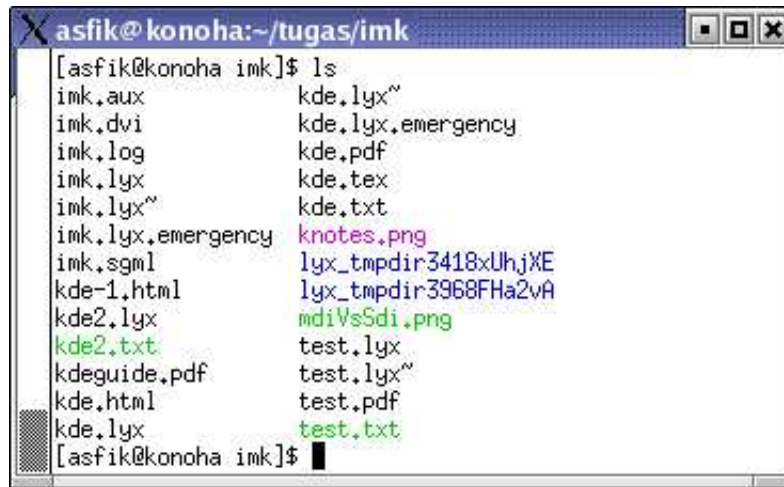


Gambar 2. MDI versus SDI.

Dalam SDI terdapat beberapa bentuk, antara lain adalah : Pure SDI, Co-Operating SDI, dan Controlled SDI.

Pure SDI

Dalam model Pure SDI, sebuah *window* aplikasi hanya mempunyai satu buah dokumen untuk semua tujuan yang akan diperoleh dari semua aplikasi didalamnya. Tidak ada *window* utama yang akan mengontrol semua *window* bagian didalamnya karena tidak saling berhubungan. Sebagai contoh perhatikan sebuah *virtual terminal window*, misalnya adalah aplikasi **xterm**. Setiap aplikasi tersebut mempunyai kontrol sendiri pada setiap *window*-nya. Menutup sebuah **xterm** tidak akan menyebabkan **xterm** yang lain menjadi tertutup.



```

[asfik@konoha imk]$ ls
imk.aux          kde.lyx~
imk.dvi          kde.lyx.emergency
imk.log          kde.pdf
imk.lyx          kde.tex
imk.lyx~         kde.txt
imk.lyx.emergency knotes.png
imk.sgml         lyx_tmpdir3418xUhjxE
kde-1.html       lyx_tmpdir3968FHa2vA
kde2.lyx         mdiVsSdi.png
kde2.txt         test.lyx
kdeguide.pdf    test.lyx~
kde.html         test.pdf
kde.lyx         test.txt
[asfik@konoha imk]$

```

Gambar 3. Aplikasi dengan model Pure SDI (xterm).

Co-Operating SDI

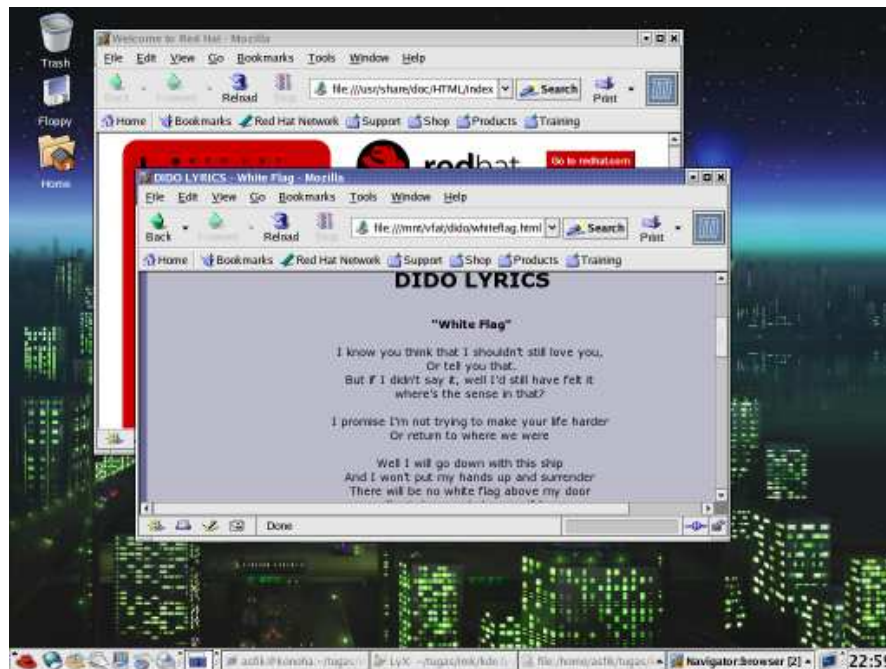
Salah satu kelemahan dari model Pure SDI adalah dalam mengontrol setting konfigurasi dari sebuah aplikasi. Perhatikan sebuah aplikasi web browser yang menggunakan model Pure SDI dimana setiap *window* sedang membuka sebuah halaman web dan terisolasi antara *window* satu dan lainnya. Model ini sebenarnya bekerja dengan baik, namun terjadi kerancuan apabila pengguna akan melakukan perubahan setting pada *bookmark*, besar kecil atau jenis *font* misalnya. Apakah perubahan *bookmark* akan diterapkan hanya pada *window* dimana hal tersebut dilakukan atau pada semua *window*? Apakah perubahan besar kecil atau jenis *font* akan diterapkan hanya pada *window* dimana hal tersebut dilakukan atau pada semua *window*?

Dalam beberapa hal penyelesaian masalah ini bisa bermacam-macam. Adalah tugas pembuat aplikasi yang menentukan hal terbaik yang harus diputuskan. Namun demikian, jika perubahan tersebut dilakukan pada semua aplikasi *window* serupa yang sedang berjalan, maka hal tersebut akan melanggar ketentuan dari model Pure SDI. Karena pada model tersebut semua aplikasi yang berjalan adalah independen, tidak tergantung antara satu dengan yang lainnya, dimana suatu perubahan pada sebuah aplikasi *window* tidak akan berakibat pada *window* yang lainnya.

Jika seorang pembuat aplikasi memutuskan untuk berinteraksi dengan hal ini, maka dia harus berpindah dari model Pure SDI dan mulai menggunakan model Co-Operation SDI. Model Co-Operating SDI merupakan sebuah model dibuat yang berdasarkan model Pure SDI yang memungkinkan pembuat aplikasi menambahkan fitur yang bisa digunakan dimana fitur tersebut tidak diijinkan

pada model Pure SDI.

Sebuah aplikasi bisa menambahkan menu tambahan pada *toolbar* utama, contohnya adalah menu Windows. Menu ini akan berisi daftar dari dokumen yang sedang dibuka, dimana jika menu ini dipilih maka akan menampilkan dokumen yang berkaitan. Pada menu Windows bisa juga ditambahkan item Close All yang terpisah dari daftar dokumen yang terbuka, sehingga jika item ini dipilih akan menutup semua dokumen yang ada beserta *window*-nya.



Gambar 4. Sebuah web browser yang menggunakan Pure SDI.

Controlled SDI

Pada model Pure SDI dan Co-Operation SDI mempunyai sebuah fitur yang memiliki masalah. Jika sebuah aplikasi membutuhkan kemampuan untuk membuka file yang lebih banyak daripada jumlah *windows* yang tersedia, maka suatu keputusan harus dibuat untuk mengontrol masalah tersebut untuk kemudian ditampilkan pada *windows* yang tersedia saja. Sebagai contoh, sebuah aplikasi dijalankan dari *command line* seperti berikut “Kapp *.jpg”, dan pada direktori dimana perintah tersebut dijalankan terdapat 2000 buah file JPEG. Pada model Pure SDI dimana sebuah dokumen mempunyai sebuah *window*, hal ini merupakan hal yang merepotkan. Penyelesaian dari masalah diatas adalah dengan membuat sebuah *window* yang berisi daftar dari semua file yang dibuka

dan siap untuk digunakan. Dokumen *window* yang baru akan dibuat ketika file tersebut dibuka. Hal ini disebut dengan Controlled SDI.

Hal yang perlu dipahami adalah bahwa *window* kontrol utama tidak membatasi pemindahan *window* yang lain. Ini bukan merupakan *interface* MDI, tetapi *interface* SDI yang menggunakan mekanisme pengontrolan. Controlled SDI dibentuk dari model Co-Operation SDI dengan beberapa fitur tambahan.

Kesimpulannya, model *windows* yang digunakan KDE berdasarkan pada SDI - Single Document Interface. Model Pure SDI sebaiknya digunakan jika memungkinkan. Namun jika tidak memungkinkan pembuat aplikasi bisa menggunakan model Co-Operation SDI, untuk aplikasi yang lebih kompleks bisa menggunakan model Controlled SDI yang digunakan untuk membuka beberapa aplikasi yang membutuhkan *windows* yang lebih dari satu.

2.1.4 Labels

Label digunakan untuk memberikan suatu keterangan pada sebuah objek atau elemen. Terdapat 2 macam bentuk penggunaan besar-kecilnya huruf pada suatu label, yaitu *booktitle* dan *sentence style*. Pada model *booktitle*, sebuah label menggunakan huruf besar pada setiap huruf awal pada sebuah kata. Perlu diperhatikan bahwa tidak semua kata harus dibuat huruf besar pada awalnya, misalnya untuk :

1. Preposisi (kata depan) yang mempunyai huruf kurang dari lima kecuali "From", seperti misalnya "for", "in", "with" atau "to"
2. Konjungsi (kata penghubung), misalnya "and", "or" dan "but".
3. Kata bantu, misalnya "the", "a" dan "an".

Contoh penggunaan *style* pada label :

- Example of Book Title Capitalization
- Example of sentence style capitalization

Penggunaan Book title capitalization biasanya dilakukan pada :

- window and dialog box titles
- group box / group line labels
- button labels
- tab labels

- listview column headers
- menu titles / menu items
- derivatives of KCommand
- combobox items
- listbox items
- tree list items
- heading/title text

Penggunaan Sentence style capitalization biasanya dilakukan pada :

- edit box labels
- list box labels
- combo box labels
- spin box labels
- check box labels
- option button labels
- spin box labels
- slider labels
- pop-up hint text
- non heading/title text

Label yang digunakan untuk sekelompok pilihan, misalnya sebuah label pada *frame* yang berisi sekelompok pilihan, **tidak** boleh menggunakan akhiran : (titik dua). Sedangkan sebaliknya, sebuah label didepan sebuah pilihan harus menggunakan akhiran : (titik dua).

2.1.5 Settings

Aplikasi-aplikasi di KDE mempunyai tiga macam tipe seting yang berbeda, yaitu : *options*, *document options* dan *configuration*. Seting-seting tersebut nantinya akan berinteraksi dengan Session Management.

- **Options**
Options dapat diubah melalui menu *setting* atau memodifikasi secara langsung terhadap elemen GUI yang berhubungan. Akibat dari perubahan ini hanya diterapkan kepada aplikasi yang aktif. Sebagai contoh, adalah letak posisi dari *toolbar*. Menyimpan seting konfigurasi ketika aplikasi tersebut ditutup adalah hal yang baik, namun itu tidak selalu mutlak.
- **Document Options**
Document Options secara umum dibatasi hanya pada suatu dokumen atau lokasi yang spesifik. Options akan disimpan ketika sebuah dokumen atau lokasi disimpan.
- **Configuration**
Konfigurasi dari suatu aplikasi bisa diubah melalui menu "Configure". Perubahan dari konfigurasi akan disimpan ketika pengguna menekan tombol OK dan harus segera diterapkan pada semua aplikasi serupa yang sedang berjalan.
- **Session Management**
Session Management adalah kemampuan KDE untuk menyimpan semua properti dari *desktop* dan *me-restore* (menampilkannya) dilain waktu. Properti dari *desktop* tersebut disebut dengan "Session". Ketika sebuah aplikasi disimpan oleh Session Management, maka aplikasi tersebut harus menyimpan *options*-nya sendiri. Aplikasi juga harus menyimpan file pada lokasi yang telah ditentukan, hal ini biasanya berhubungan dengan pembukaan sebuah dokumen.

2.1.6 Systray

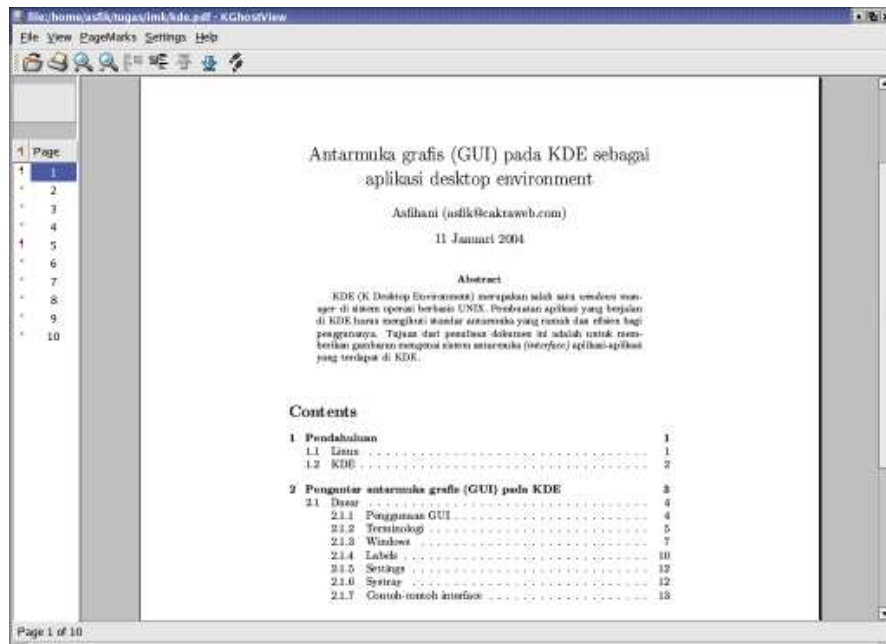
Pada KDE versi 3 keatas dikenalkan *systray*. Systray adalah sebuah entri berupa *icon* yang tampak pada *system tray* (biasanya dipanel) yang merepresentasikan sebuah aplikasi. Menu pada *systray* bisa berupa *popup menu* sederhana sampai yang rumit dan kompleks.

Operasi yang biasa dilakukan pada sebuah *systray* adalah :

- **Klik tunggal**
Menampilkan/menyembunyikan *interface* aplikasi utama.
- **Klik kanan**
Menampilkan menu yang berisi opsi keluar dari aplikasi atau melakukan konfigurasi.

2.1.7 Interface example

Berikut ini adalah contoh sebuah *user interface* pada sebuah aplikasi untuk melihat dokumen PDF. Perhatikan urutan menu dan desain dari *toolbar*, *statusbar* dan *content area* (tempat kerja/isi).

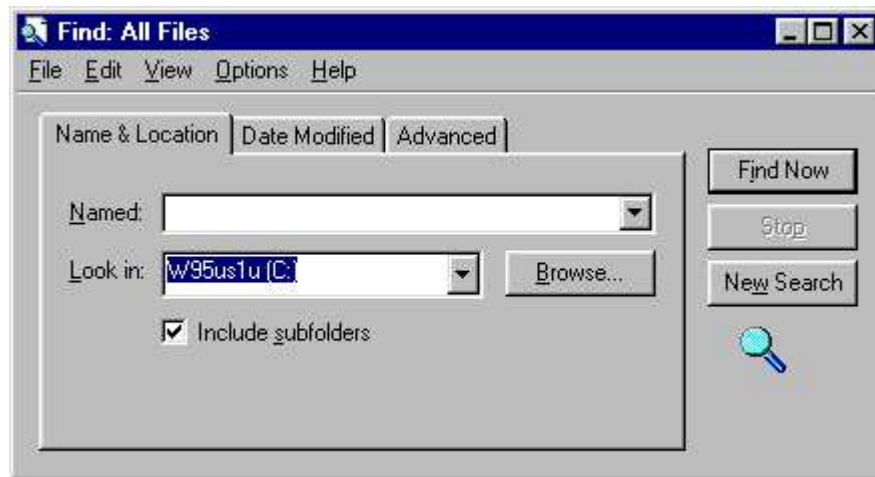


Gambar 5. Contoh aplikasi pembaca dokumen PDF (kghostview).

Beberapa aplikasi, biasanya sebuah aplikasi *utility* sederhana, tidak mematuhi menggunakan kaidah *user interface*. Aplikasi tersebut seharusnya menggunakan sebuah *dialog box*. Namun pada penerapannya, aplikasi tersebut merupakan penggabungan antara *dialog box* dengan sebuah *menu interface* yang akhirnya sering menimbulkan kerancuan. Kesimpulannya adalah hindari penggunaan *dialog box* dengan *user interface* yang berhubungan dengan suatu dokumen.

Sebagai contoh adalah applet Find di Microsoft Windows 95⁵:

⁵<http://digilander.libero.it/chiediloapippo/Engineering/iarchitect/expfind.gif>



Gambar 6. Find di Windows 95 yang tidak mematuhi kaidah user interface.

Applet Find diatas merupakan aplikasi ganjil dan aneh yang merupakan hasil penggabungan antara *dialog box* dengan *menubar*. Pada dasarnya sebuah *dialog box* tidak mempunyai menu. Menu biasanya hanya digunakan pada aplikasi yang berhubungan dengan dokumen, misalnya *text editor* ataupun aplikasi grafis. Masalah yang paling mendasar adalah bahwa Find mencoba menampilkan fungsi-fungsi yang berbeda dalam satu window secara bersamaan.

2.2 Menu

Menu menyediakan *actions* (tindakan-tindakan) yang dimungkinkan untuk dilakukan terhadap suatu dokumen. Yang perlu diperhatikan adalah tindakan yang **mungkin dilakukan**. Jika tindakan tidak mungkin untuk dilakukan (misalnya menu Cut jika tidak ada item yang dipilih), maka menu tersebut harus dinon-aktifkan. Setiap aplikasi tidak harus mengikuti semua menu yang akan dijelaskan berikut ini. Meskipun demikian, jika aplikasi tersebut menggunakan menu, maka harus sesuai dengan petunjuk berikut ini. Jika dibutuhkan, pembuat aplikasi bisa menambahkan menu tambahan yang ditunjukkan dengan APS (Application Specific Menus) agar sesuai dengan kebutuhan aplikasi tersebut.

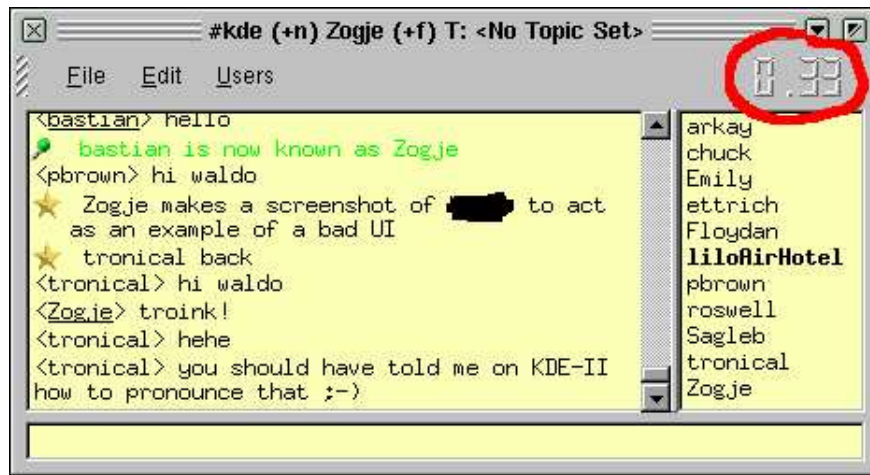
Urutan dari menu sebaiknya seperti ini, namun pembuat aplikasi bisa menghilangkan salah satu menu jika perlu :

File Edit View Go APS Bookmark Tools Setting Help

Yang perlu diperhatikan adalah bahwa menu yang akan membuka sebuah *dialog*, maka harus diberi label dengan akhiran ... (misalnya Save As..., Open...).

Pada *menubar* sebaiknya tidak menambahkan hal yang lain selain daftar menu itu sendiri.

Contoh :



Gambar 7. Jangan menampilkan informasi sesuatu pada menubar.

Item pada menu sebaiknya juga tidak dihilangkan atau ditambahkan ketika aplikasi sedang dijalankan. Akan lebih baik jika menu tersebut diaktifkan atau dinon-aktifkan, kecuali untuk item-item yang dimasukkan dalam sebuah menu, misalnya daftar *bookmarks* atau daftar website yang pernah dikunjungi. Daftar-daftar tersebut harus bisa dihilangkan atau ditambah agar bisa selalu *up-to-date*.

2.2.1 File

Isi dari menu File sangat tergantung dari aplikasi yang akan dibuat. Kategori aplikasi berdasarkan menu bisa dibedakan menjadi tiga macam, yaitu :

1. Aplikasi yang berorientasi dokumen seperti *text editors*, *word processors*, dan program manipulasi grafis.



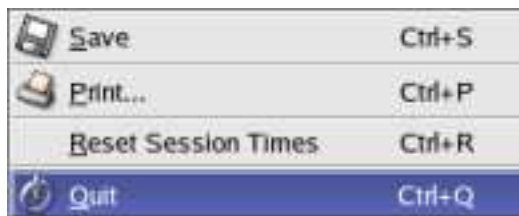
Gambar 8. Menu File pada OpenOffice.

2. Aplikasi permainan (games) atau hiburan.



Gambar 9. Menu pada Kolf.

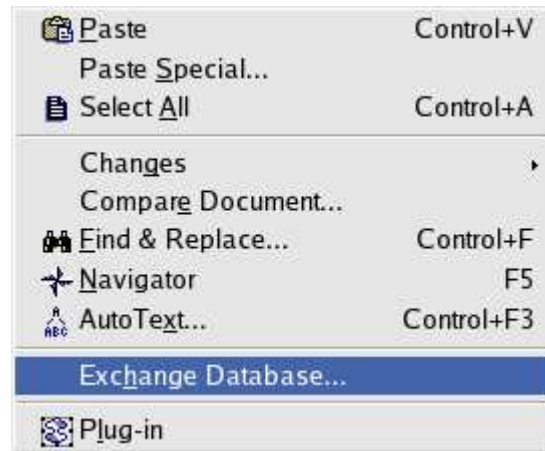
3. Aplikasi yang lainnya.



Gambar 10. Menu File pada KArm.

2.2.2 Edit

Menu Edit hanya akan memanipulasi objek atau elemen yang terdapat di *content area*, tidak ada fungsi yang lain.



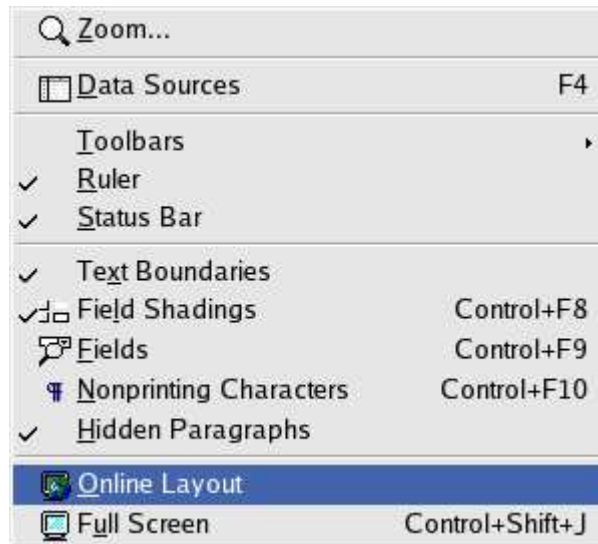
Gambar 11. Menu Edit pada OpenOffice.

Pembuat aplikasi harus memperhatikan bahwa :

- Non-aktifkan Undo jika tidak ada sesuatu yang bisa di-*undo*.
- Non-aktifkan Redo jika tidak ada sesuatu yang bisa di-*redo*.
- Non-aktifkan Cut dan Copy jika tidak ada sesuatu yang dipilih (diblok).
- Non-aktifkan Paste jika sebelumnya tidak ada sesuatu yang di-*cut* or di-*copy*.
- Non-aktifkan Find Next sampai Find dieksekusi terlebih dahulu.

2.2.3 View

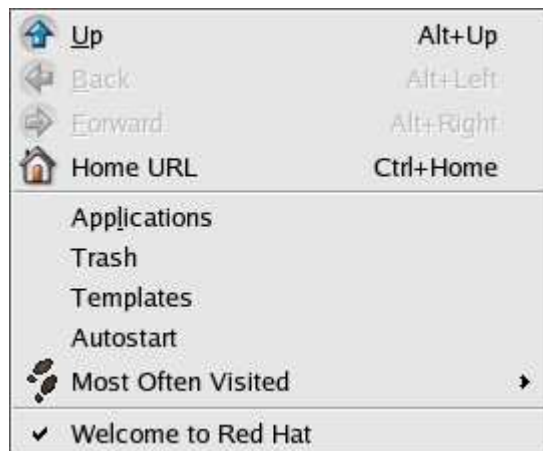
Semua *actions* yang tersedia pada menu View memanipulasi tampilan dari *content area*, namun tidak akan memanipulasi objek atau elemen didalam *content area* itu sendiri.



Gambar 12. Menu View pada OpenOffice.

2.2.4 Go

Tergantung dari aplikasi, sebuah aplikasi bisa mempunyai web-style *Go menu* atau sebuah *document Go menu*. Atau jika tidak diperlukan, maka tidak perlu mempunyai menu ini.



Gambar 13. Menu Go pada Konqueror.

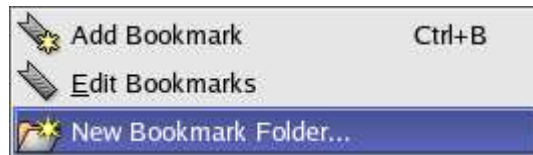
Pembuat aplikasi harus memperhatikan bahwa :

- Non-aktifkan Back jika berada pada halaman pertama.

- Non-aktifkan Forward jika berada pada halaman terakhir.

2.2.5 Bookmarks

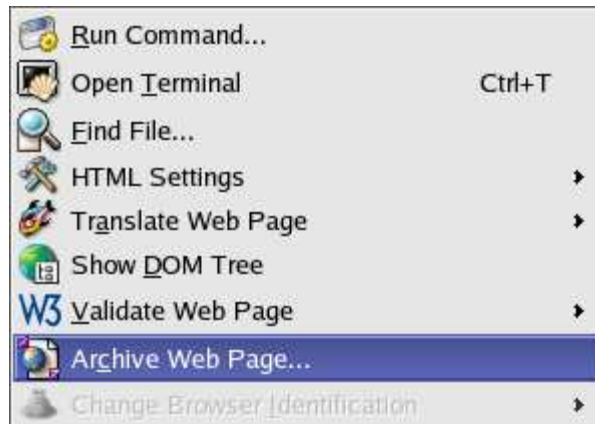
Menu Bookmarks digunakan untuk menyimpan alamat lokasi dokumen atau URL yang sering digunakan atau diakses. Namun, menu ini tergantung dari jenis aplikasi yang akan dibuat.



Gambar 14. Menu Bookmark pada Konqueror.

2.2.6 Tools

Menu Tools berisi *actions* atau alat bantu yang lain yang berfungsi untuk melakukan manipulasi elemen atau fungsi spesifik yang lainnya.



Gambar 15. Menu Tools pada Konqueror.

2.2.7 Settings

Menu Settings digunakan untuk mengatur bagaimana sebuah aplikasi akan digunakan, serta merupakan kontrol terhadap elemen aplikasi yang lainnya, misalnya kontrol terhadap *menubar*.

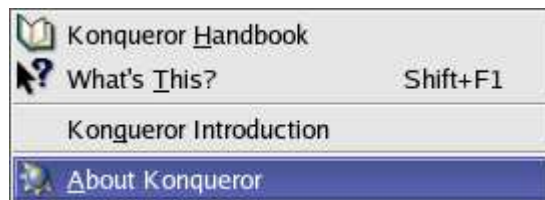


Gambar 16. Menu Settings pada Konqueror.

Jika pengguna diijinkan untuk menyembunyikan *menubar*, maka harus dipastikan bahwa untuk menampilkan kembali *menubar* yang tersembunyi tersebut dengan mudah.

2.2.8 Help

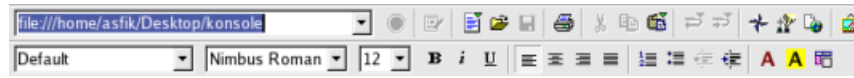
Menu Help berisi bagaimana cara menggunakan aplikasi yang berkaitan, lisensi, maupun informasi tentang pembuat aplikasi.



Gambar 17. Menu Help pada Konqueror.

2.3 Toolbar

Pada aplikasi terdahulu, biasanya *toolbar* yang digunakan adalah *buttonbar*, di mana terdapat sebuah *bar* yang berisi tombol-tombol yang digunakan untuk menjalankan fungsi atau tindakan tertentu. KDE *toolbars* bisa menggunakan *buttonbar* atau *widgets* yang lain, misalnya: *labels*, *line edits*, *combo boxes*, dan *frames* (yang dapat berisi kontrol atau elemen yang lain). Fungsi dari *toolbar* adalah untuk menawarkan *actions* cepat dan sering digunakan yang bisa dijalankan melalui sebuah kontrol. Yang perlu diperhatikan adalah, semua *actions* tersebut harus bisa dijalankan melalui *menubar*.



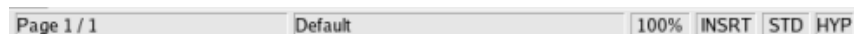
Gambar 18. Toolbar pada OpenOffice.

Tidak perlu untuk memasukkan semua fungsi pada *toolbar*. Namun demikian fungsi-fungsi berikut ini harus ditampilkan di *buttonbar* jika memang dibutuhkan:

- New
- Open
- Save
- Print
- Print Preview
- Undo
- Redo
- Cut
- Copy
- Paste
- Find
- Zoom
- Previous Page/Back
- Next Page/Forward
- Go To Page/Home
- Help

2.4 Statusbar

Statusbar digunakan untuk menampilkan sesuatu yang terjadi pada dokumen yang sedang diolah. Biasanya terdapat beberapa *field* informasi pada *bar*-nya.



Gambar 19. Statusbar pada OpenOffice.

2.5 Content Area

Content area berisi segala sesuatu yang ditampilkan pada *window* utama dari sebuah aplikasi, misalnya sebuah dokumen. Pada *content area* bisa juga ditambahkan elemen *user interface* yang lain, misalnya *tree lists* dan sebagainya.



Gambar 20. Content area pada OpenOffice.

2.6 Dialogs

Sebuah *dialog* digunakan sebagai salah satu cara berinteraksi antara pengguna dan aplikasi. Dialog akan memberikan beberapa pilihan kepada pengguna tentang tindakan yang akan dilakukan.

Dialogs dikategorikan menjadi 3 macam :

1. Simple Dialogs

Suatu *dialogs* yang akan menampilkan sebuah pertanyaan, termasuk pertanyaan Yes-No.

2. Complex Dialogs

Suatu *dialogs* yang memungkinkan pengguna untuk mengoptimalkan suatu operasi.

3. Standar Dialogs

Suatu *complex dialogs* yang disediakan KDE untuk operasi yang sering digunakan seperti misalnya mengeprint dan pemilihan terhadap suatu file.

Semua *dialogs* harus mempunyai judul yang sesuai (berkaitan). Didalam *dialogs* tidak diijinkan menggunakan *menubar* maupun *toolbar*.

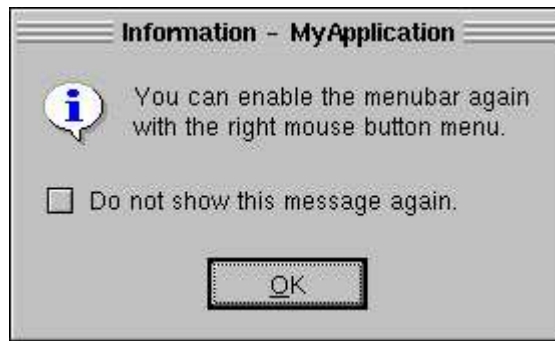
2.6.1 Simple Dialogs

Simple dialogs hanya menampilkan sebuah pertanyaan sederhana kepada penggunanya. Simple dialogs bisa dikategorikan menjadi 4 jenis, yaitu:

1. Notifications

Notifications adalah sebuah *message* yang dikirim ke pengguna, dimana tidak ada pilihan yang ditawarkan. Ketika pengguna sudah selesai, maka tinggal menekan tombol **OK** untuk menghilangkan notifikasi. Penekanan tombol *Escape* atau *Enter* akan melakukan hal yang sama.

Contoh :

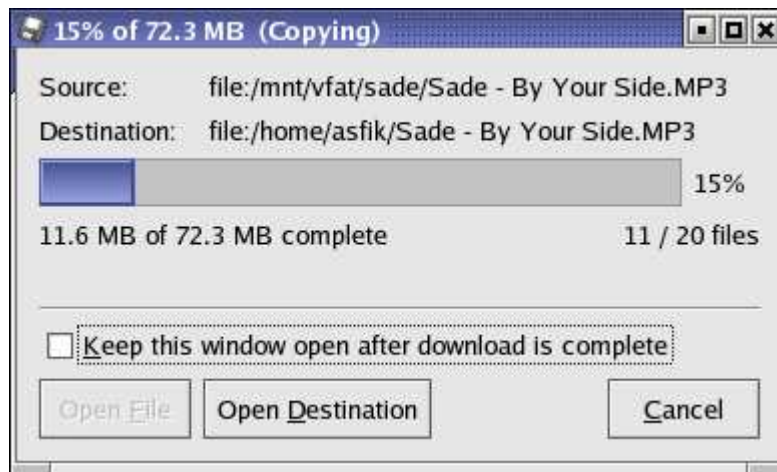


Gambar 21. Notifikasi pada sebuah aplikasi.

2. Progress Dialogs

Progress Dialogs merupakan *dialogs* yang digunakan untuk menampilkan aktifitas yang sedang berlangsung. Pengguna harus diberi pilihan untuk membatalkan aktifitas tersebut. Pilihan dengan label **Cancel** akan dilaksanakan jika hasil yang dikerluarkan sama dengan tidak melakukan perubahan, sedangkan pilihan dengan label **Stop** akan menghentikan proses dengan hasil parsial.

Contoh :

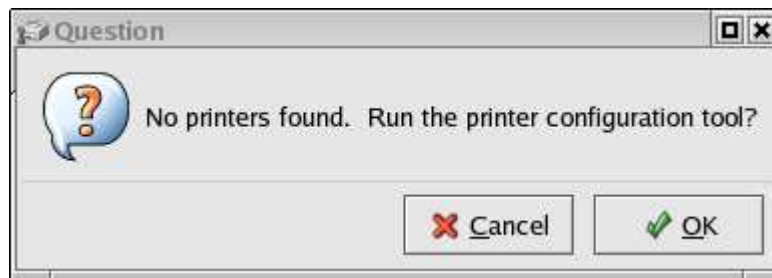


Gambar 22. Progress bar pada proses penyalinan file.

3. Yes-No Questions

Yes-No Questions merupakan sebuah *dialogs* yang menampilkan dua pilihan sederhana yaitu **Yes** dan **No**. Pilihan Yes akan melaksanakan *action* yang sesuai, kebalikannya pilihan No akan membatalkannya. Namun demikian, label yang lain juga bisa dipakai, tidak harus label Yes dan No.

Contoh :

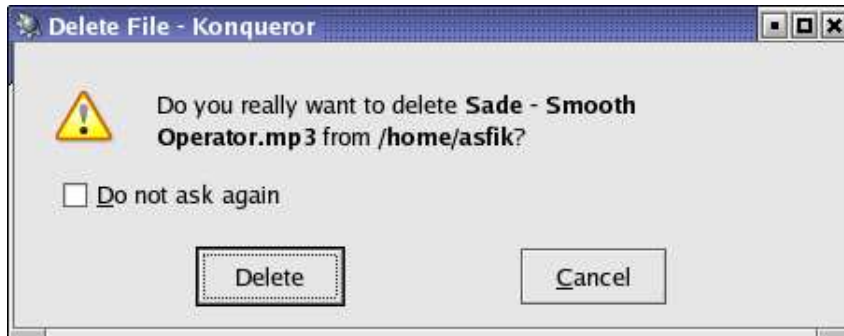


Gambar 23. Yes-No Question pada sebuah aplikasi.

4. Confirmation Requests

Confirmation Requests merupakan sebuah *dialogs* yang akan memberikan informasi kepastian kepada pengguna tentang *action* apa yang akan dikerjakan. Pengguna mempunyai dua pilihan yaitu tombol **OK** dan tombol **Cancel**. Label OK yang digunakan disarankan disesuaikan dengan *action* yang akan dilakukan, misalnya pemberian label **Delete** untuk suatu operasi penghapusan (terhadap suatu file).

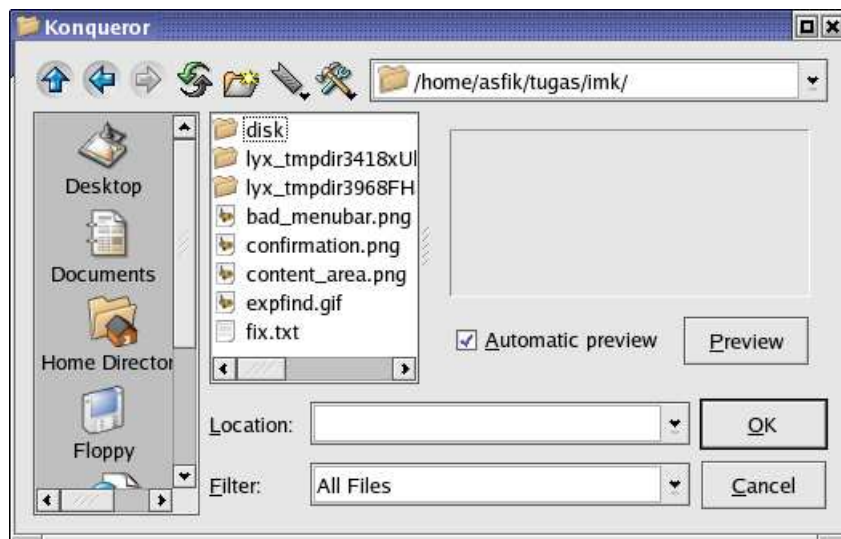
Contoh :



Gambar 24. Confirmation Request pada aplikasi Konqueror.

2.6.2 Complex Dialogs

Complex Dialogs adalah sebuah *dialogs* yang mengizinkan pengguna sebuah aplikasi untuk melakukan suatu operasi yang kompleks. Di lain hal, pilihan submenu pada sebuah menu terkadang memerlukan *argument* dan *parameter* tambahan, maka digunakanlah *complex dialogs* untuk mengatasinya. Sebagai contoh yang lain adalah *dialogs* untuk menyeleksi dan memilih suatu dokumen yang akan dibuka.



Gambar 25. Complex Dialogs pada aplikasi Konqueror.

2.6.3 Standard Dialogs

Standard Dialogs adalah *dialogs* yang disediakan oleh KDE untuk *tasks* (tugas-tugas) tertentu pada sebuah aplikasi. Standard Dialogs mencakup antara lain pada : *About Dialog*, *Color Selection Dialog*, *File Selection Dialog*, *Font Selection Dialog*, *Icon Selection Dialog*, *Key Configuration Dialog*, *Text Edit Dialogs* dan sebagainya.



Gambar 26. Standar Dialogs About pada sebuah aplikasi (Konqueror).

2.7 Keys

Keys merupakan jalan pintas pada sebuah aplikasi dengan mengkombinasikan *alphanumeric key* dengan Meta key. KDE menggunakan Accelerator keys dan Shortcut keys.

2.7.1 Accelerators

Accelerators keys kombinasi antara *alphanumeric key* dan Meta key (biasanya **Alt**) yang digunakan untuk memilih item yang kelihatan pada layar. Jika sebuah item mempunyai *accelerator key* maka akan diberi garis bawah pada karakter yang bersangkutan.

Contoh :

Alt-A adalah *accelerator key* untuk Select All.



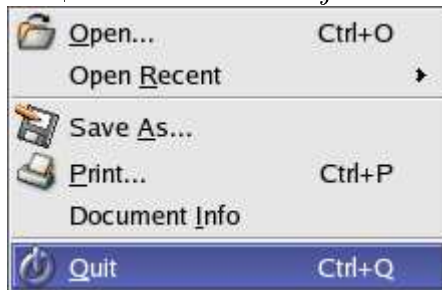
Gambar 27. Accelerator key pada menu Edit di Netscape Navigator.

2.7.2 Shortcuts

Shortcuts key adalah kombinasi antara **Ctrl** dan key yang lain. Ketika ditekan bersamaan, maka akan melakukan *action* dari menu yang berkaitan. Pada menu *shortcuts* diletakkan setelah entri pada suatu item submenu. Hal ini dimaksudkan agar pengguna bisa mengingat *shortcuts* dengan menggunakan aplikasi.

Contoh :

Ctrl+O adalah *shortcut key* untuk membuka sebuah dokumen.



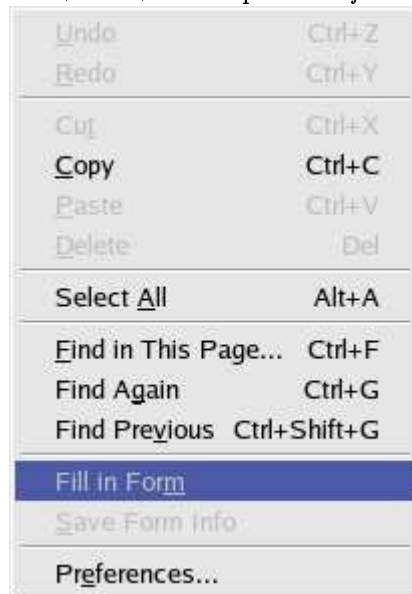
Gambar 28. Shortcut key pada menu File di KGhostView.

2.7.3 Shift Modifier

Shortcuts harus selalu menggunakan kombinasi **Ctrl** dan key tunggal (*single function key*). Namun jika fungsi dengan *shortcuts* tersebut sudah digunakan maka dengan menambahkan key **Shift** maka didapatkan fungsi yang sama atau kebalikannya.

Contoh :

Ctrl+Shift+G merupakan *shift modifier* untuk Find Previous.



Gambar 29. Shift modifier pada menu Edit di Netscape Navigator.

2.7.4 Cursor

Pada aplikasi-aplikasi *editor* dan *word-processor*, tanda **caret** (atau *cursor*) merupakan tanda dimana suatu *command editing* akan diterapkan. Cursor keys digunakan misalnya untuk memindahkan *caret*, menggulung grafis, memilih suatu daftar item, dan lain sebagainya.

Contoh :

Key	Pergerakan	Pergerakan Ctrl ditekan
Left	Bergerak satu unit kekiri, mis: karakter	Bergerak kekiri dalam unit besar, mis: kata
Right	Bergerak satu unit kekanan	Bergerak kekanan dalam unit besar

2.7.5 Selection

Terkadang pengguna mouse untuk memilih (mengeblok) suatu teks pada mode *editing* merepotkan. Sebenarnya, teks bisa dipilih dengan menggunakan kombinasi *cursor keys* dengan key **Shift** yang ditekan.

Contoh :

Key	Pemilihan	Pemilihan dengan Shift ditekan
Shift+Left	Memilih satu karakter ke kiri	Memilih sampai ke awal kata ke kiri
Shift+Right	Memilih satu karakter ke kanan	Memilih sampai ke awal kata ke kanan
Shift+Up	Memilih satu baris ke atas	Memilih sampai ke awal paragraf

2.7.6 Dialog Navigation

Pengguna tidak harus selalu menggunakan mouse untuk menggunakan aplikasi. Kenyataannya, menggunakan keyboard untuk berpindah-pindah pilihan pada sebuah *dialog* lebih efisien karena pengguna tidak perlu untuk menggeser-geser sebuah mouse.

Contoh :

Key	Fokus Pergerakan
Tab	Berpindah ke field berikutnya
Shift+Tab	Pindah ke field sebelumnya

2.7.7 Completion

Ketika pengguna mengisi suatu isian pada sebuah *user interface*, suatu aplikasi harus bisa menawarkan informasi yang kira-kira bisa diberikan. Sebagai contoh, ketika pengguna mengisikan beberapa buah karakter suatu alamat email, maka suatu aplikasi akan menawarkan alamat email yang lengkap yang didapatkan dari alamat email yang telah dikirimkan atau dari *address book*.



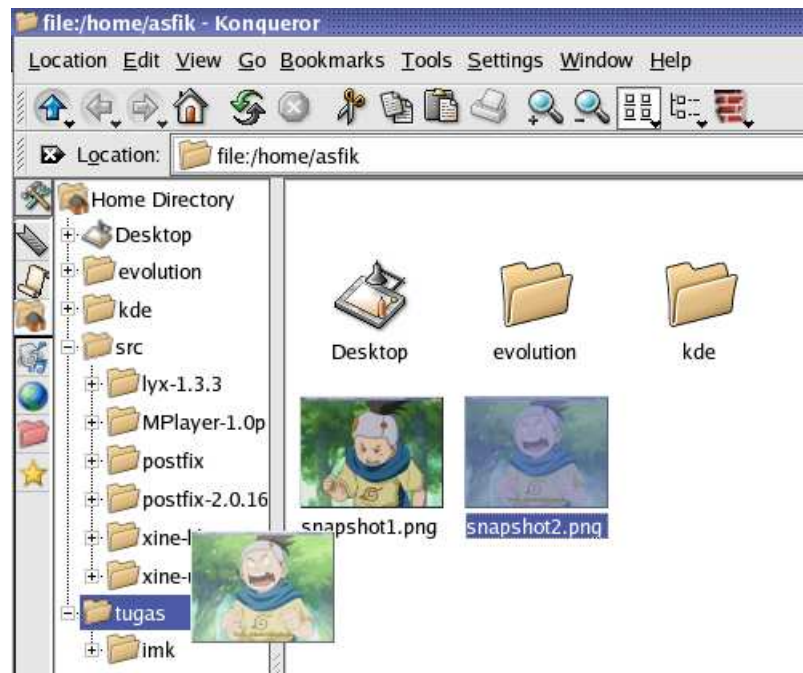
Gambar 30. Auto Completion pada aplikasi Ximian.

2.8 Mouse

Pada sebuah aplikasi, pada umumnya sebuah mouse digunakan sebagai alat penunjuk untuk memilih suatu objek atau menu serta menjalankan fungsi-fungsi yang lain. Di KDE, objek tersebut bisa sebuah file, *links*, ataupun *buttons* (tombol). Penekanan bisa dilakukan dengan *single-click* (penekanan tunggal) atau *double-click* (penekanan ganda). Biasanya, jika tombol kiri dari mouse diklik, maka akan melakukan *action*, namun jika tombol kanan diklik, maka harus ditampilkan sebuah *pop up* menu.

2.9 Drag and Drop

Drag and Drop memungkinkan pengguna aplikasi untuk melakukan *drag* (penyeretan) sebuah objek dan menge-*drop*-nya (melepaskan) pada tempat lain. Pengguna melakukan *drag* dengan memilih suatu objek (misalnya file), kemudian dengan mouse masih ditekan, menggesernya ke lokasi yang lain. Ketika pengguna melepaskan tekanan mouse, maka objek akan *drop* (dilepaskan). Drag and Drop akan efektif jika dilakukan pada sebuah *window* aplikasi yang mempunyai bagian-bagian terlihat dari sisi pengguna, sehingga hal ini akan memudahkannya.



Gambar 31. Drag and Drop pada suatu aplikasi (Konqueror).

3 Daftar Pustaka

- KDE Official Website, <http://www.kde.org>
- KDE User Interface Guidelines, <http://developer.kde.org>
- Interface Hall of Shame,
<http://digilander.libero.it/chiediloapippo/Engineering/iarchitect/shame.htm>