



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

TINJAUAN PUSTAKA

2.1. Pertolongan Pertama Pada Kecelakaan

Kecelakaan dapat terjadi dimana saja, kapan saja dan pada siapapun. Jika tidak ditangan secara serius, kecelakaan dapat berakibat fatal. Sebelum menerima pertolongan lebih lanjut, penderita yang sedang berada dalam keadaan darurat dapat menerima pertolongan pertama terlebih dahulu untuk mencegah situasi menjadi lebih parah. Menurut Wulandari (2014), kecelakaan merupakan sebuah kejadian yang dapat mengakibatkan seseorang mengalami sakit atau cedera. Faktor penyebab dari terjadinya kecelakaan bisa berasal dari kondisi maupun perbuatan berbahaya. Kejadian-kejadian yang dianggap sebagai kecelakaan adalah kejadian yang tidak terduga dan dapat menimbulkan kerugian dan penderitaan (hlm. 11).

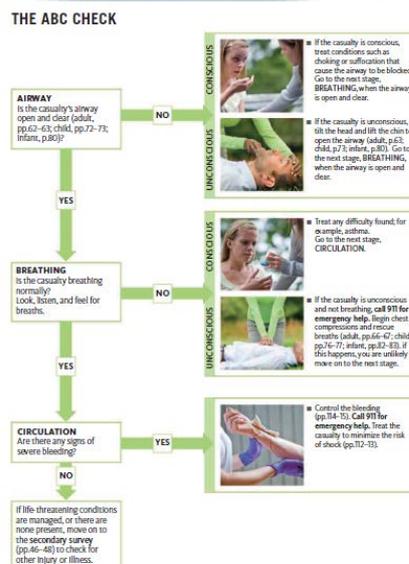
The American College of Emergency Physicians (2014) menjelaskan bahwa pertolongan pertama merupakan tindakan paling awal dalam menanggapi seseorang yang terluka atau mendadak jatuh sakit. Selain bertujuan untuk menyelamatkan korban, pertolongan pertama juga dilakukan untuk memastikan kalau semua orang yang terlibat tetap dalam kondisi aman dan menjaga supaya keadaan tidak menjadi semakin parah (hlm. 12-14).

Dalam saat darurat, penting bagi penolong untuk mengikuti ketiga langkah ini sebelum memberikan pertolongan pertama yaitu mengecek situasi sekeliling, memastikan lokasi kejadian aman dan yang terakhir baru memberikan pertolongan

pertama (*The American College of Emergency Physicians*, 2014, hlm. 28). Pada pertolongan pertama, terdapat tiga aspek penting yang harus diperhatikan dalam menangani korban. Ketiga aspek tersebut berdasarkan urutan yang paling penting sebagai berikut (hlm. 40):

1. Memeriksa apa yang terjadi pada korban.
2. Lakukan penanganan terhadap cedera yang paling parah terlebih dahulu.
3. Memutuskan penanganan lebih lanjut yang diperlukan oleh korban. Bila perlu, hubungi bantuan darurat.

Apabila dalam sebuah kecelakaan terdapat lebih dari satu korban yang mengalami cedera parah, penolong harus menjalani *primary survey* untuk menentukan korban mana yang harus ditolong terlebih dahulu. Dalam menjalankan *primary survey*, penolong dapat lebih mudah mengingat grafik *ABC check*, dimana ABC merupakan singkatan dari *airway*, *breathing* dan *circulation*.



Gambar 2.1. Bagan ABC Check
(First Aid Manual: The Step-by-step Guide for Everyone, 2014)

Setelah menjalankan *primary survey*, penolong baru dapat melanjutkan ke langkah selanjutnya yaitu *secondary survey*. *Secondary survey* terdiri dari *disability* dan *examine the casualty*. Tujuan proses peninjauan tersebut adalah untuk mencari tahu asal usul terjadinya kecelakaan tersebut, gejala dan tanda-tanda yang ditemukan pada penderita (*history, symptoms, signs*) (*The American College of Emergency Physicians*, 2014, hlm. 44-47).

Dalam pemilihan jenis pertolongan pertama yang akan disediakan dalam aplikasi tidak terdapat batasan tertentu. Kontak bantuan terdekat dan tercepat berupa ambulans dan rumah sakit terdekat juga disediakan dalam aplikasi untuk mendapat penanganan lebih lanjut segera.

2.2. Media Interaktif

O' Neill (2008) mengeksplorasi arti kata 'media' sendiri dari kata dasar '*medium*' dan berkonklusi bahwa pemahaman dari 'media' merupakan cara mengekspresikan diri sendiri atau berkomunikasi dengan sesama. Media berperan sebagai perantara diantara masyarakat satu dengan yang lain (hlm. 9). Media memiliki berbagai macam bentuk, salah satunya adalah media interaktif.

Benyon (2010) menekankan bahwa dalam merancang sebuah sistem yang interaktif, salah satu pertimbangan yang paling penting adalah fokus dimana sistem yang bersifat *human-centred* (hlm. 26). Perancangan sebuah sistem mementingkan perspektif yang berpusat pada manusia. Terdapat beberapa prinsip yang dapat menjadi pegangan dalam perancangan sebuah sistem yang *human-*

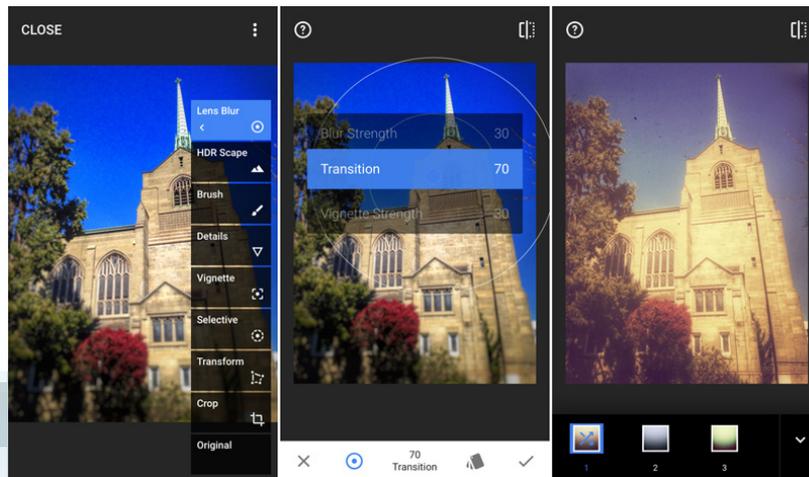
centred, yaitu *visibility*, *consistency*, *familiarity*, *affordance*, *navigation*, *control*, *feedback*, *recovery*, *constraints*, *flexibility*, *style* dan *conviviality* (hlm. 90).

Media interaktif memiliki berbagai macam bentuk, salah satu bentuk dari media interaktif adalah *mobile application*. Menurut Unhelkar (2009), sebuah *mobile application* adalah sebuah program komputer yang dirancang untuk dapat berfungsi pada perangkat *mobile* dan dapat menyajikan berbagai macam layanan untuk para pengguna. Di beberapa daerah, sebuah *mobile application* dapat menawarkan *information support* (hlm. 161).

Mobile application tidak hanya memiliki satu jenis, melainkan terbagi dalam beberapa kategori. Menurut Budi (2013, diakses pada 5 Maret 2016), berikut adalah beberapa jenis *mobile application* yang paling umum:

1. *Native Application*

Native application adalah aplikasi yang diunduh melalui *application store*, seperti *Google Play* atau *App Store*. Jenis aplikasi ini memiliki tingkat kinerja yang tinggi dan juga sangat handal. *Native mobile apps* memiliki akses penuh ke berbagai perangkat telepon, seperti kamera, GPS, dan lain-lain. Selain itu, pengguna dapat menggunakan beberapa aplikasi tanpa menggunakan koneksi internet. Namun, biaya untuk mengembangkan jenis aplikasi ini mahal karena aplikasi ini terikat kepada satu jenis *operating system*, sehingga menyebabkan para perusahaan terpaksa untuk membuat duplikat aplikasi tersebut yang dapat bekerja pada *platform* lainnya. Aplikasi jenis ini sama sekali tidak memiliki tampilan *web*, salah satu contoh *native application* adalah aplikasi *photo editor*, *Snapseed*.



Gambar 2.2. Aplikasi *Snapseed*
 (<http://www.droid-life.com/tag/snapseed/>, 2016)

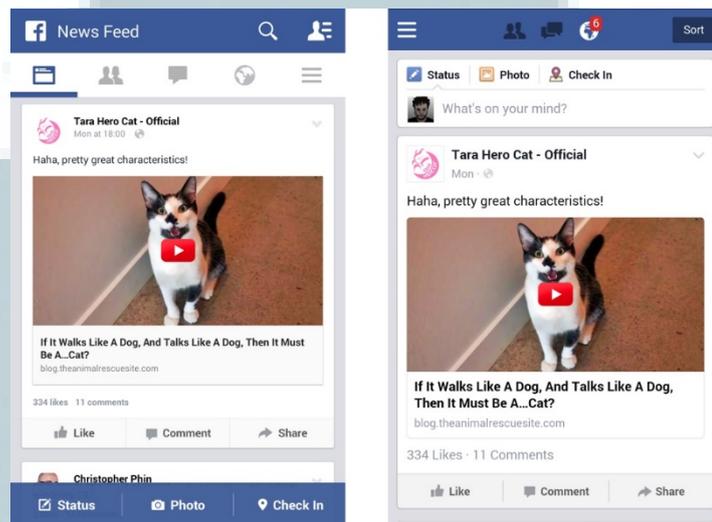
2. *Web Application*

Web application sebenarnya bukanlah sebuah aplikasi, melainkan sebuah *website* yang memiliki tampilan dan berperilaku seperti layaknya sebuah aplikasi. Aplikasi jenis ini terlihat seperti *native application*, tapi tidak diimplementasikan sedemikian. *Web application* dapat diakses dengan menggunakan *browser* dan umumnya tertulis dalam format HTML5.

3. *Hybrid Application*

Hybrid Application merupakan penggabungan dari *native application* dan *web application*. Aplikasi ini terdapat dalam *app store* dan menguasai banyak fitur perangkat yang tersedia, sama seperti *native application*. Di sisi lain, aplikasi ini bergantung pada HTML, sama seperti *web application*. Perusahaan-perusahaan sering menjadikan *hybrid application* sebagai pembungkus untuk halaman web yang ada dengan harapan untuk dapat muncul di *app store*, tanpa menghabiskan upaya signifikan untuk mengembangkan aplikasi yang berbeda. Aplikasi ini juga populer karena memungkinkan pengembangan *cross-platform*, yaitu dengan

menggunakan komponen kode HTML yang sama, dapat digunakan kembali pada *mobile operating system* berbeda yang sekaligus mengurangi biaya pengembangan. Contoh *hybrid application* yang tepat adalah *Facebook*.



Gambar 2.3. Aplikasi *Facebook*

(http://ignorethecode.net/blog/2014/11/04/web_apps/, 2014)

Dalam pembuatan *mobile application*, diperlukan beberapa elemen-elemen tertentu. Salah satu elemen yang paling mendasar adalah *mobile platform*, dalam kata lain lebih dikenal sebagai *operating system*. Saat ini, *operating system* yang paling umum digunakan adalah *Android* yang diluncurkan oleh *Google* dan *iOs* yang eksklusif digunakan hanya untuk produk *Apple*.

Berdasarkan penjelasan mengenai *mobile application* diatas, aplikasi pertolongan pertama yang dirancang termasuk kategori *native application* dimana aplikasi tersebut dapat ditemukan di *app store* serta memiliki akses penuh ke berbagai perangkat telepon. Selain itu, perancangan aplikasi tersebut hanya dapat digunakan terbatas dengan satu *operating system*, yang dalam batasan

perancangan aplikasi ini yaitu *Android*. Dalam merancang aplikasi yang efektif diperlukan desain *interface* yang *user-friendly* dengan pertimbangan menggunakan gaya desain yang konsisten, struktur navigasi yang jelas dan konten yang informatif serta mudah dimengerti.

2.3. Desain Interaktif

Pada dasarnya, sebuah desain merupakan sebuah rancangan. Mendesain berarti merencanakan atau mengorganisir. Pengertian desain tidak jauh dari ilmu seni mulai dari seni lukis, gambar, patung, fotografi dan media-media lainnya. Menurut Graham (1999) desain interaktif merupakan sebuah dokumen interaktif hasil dari gabungan berbagai macam media, seperti teks, video, foto, ilustrasi, suara, animasi, dan sebagainya (hlm.4).

Prinsip-prinsip dalam merancang desain interaktif berlaku sebagai pegangan supaya dapat menghasilkan produk yang mendukung kebutuhan dan tujuan pengguna. Prinsip desain yang interaktif dapat dibedakan menjadi tiga kategori yaitu *conceptual principles*, *behavioral principles* dan *interface-level principles*. *Conceptual principles* membantu menentukan seperti apa hasil akhir produk digital tersebut. *Behavioral principles* menggambarkan bagaimana sebuah produk bereaksi dalam konteks umum dan spesifik. *Interface-level principles* menggambarkan strategi yang efektif untuk pengaturan, navigasi dan komunikasi perilaku dan informasi (Cooper, Reimann., Cronin, & Noessel, C., 2014, hlm. 173).

McKay (2013) menyimpulkan bahwa dalam mendesain tampilan visual *mobile interface* supaya dapat berkomunikasi dengan pengguna secara efektif, hal penting yang perlu diingat adalah *user interface form follows communication* dimana fungsi diutamakan terlebih dahulu. Tata letak elemen-elemen dalam sebuah halaman diatur berdasarkan *behaviour* pengguna yang cenderung melakukan *scanning*, bukan membaca, sehingga sebisa mungkin elemen yang tidak penting atau tidak memiliki fungsi tertentu dihilangkan supaya dapat mengkomunikasikan tujuan utama dari isi halaman tersebut dengan efektif. Sebuah halaman yang didesain dengan baik terlihat sederhana, teratur, *easy to scan*, efisien dan *aesthetically pleasing* (hlm. 193).

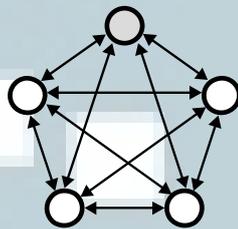
2.3.1. Navigasi

Salah satu elemen yang paling penting dalam sebuah sistem adalah navigasi. Struktur navigasi yang baik membuat pengguna dapat menelusuri sistem dengan mudah. Graham (1999) menyimpulkan bahwa navigasi merupakan proses pada saat pengguna menjelajahi segala tingkat interaktifitas melalui layar *interface*. Langkah penting dalam menciptakan skema navigasi yang baik adalah dengan melakukan analisis secara mendalam.

Salah satu cara untuk membuat skema navigasi yang baik adalah dengan melakukan visualisasi bagaimana pengguna nantinya akan menjelajahi sistem interaktif tersebut. Sebuah sistem navigasi yang dirancang dengan baik akan memperjelas isi dan struktur navigasi, meningkatkan kegunaan dokumen tersebut dan membantu memenuhi kebutuhan pengguna (hlm. 51-53).

Struktur navigasi memiliki beberapa model yang berbeda. Tidwell (2010) menyatakan bahwa beberapa *navigational model* yang sering ditemukan dalam *website* maupun aplikasi adalah (hlm. 80-84):

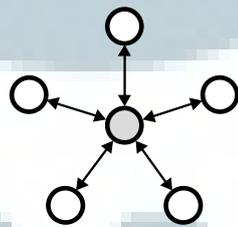
1. *Hub and spoke*: pengguna memulai dari *hub*, mengklik dan menuju *spoke* lalu kembali ke *hub* jika ingin melanjutkan ke *spoke* lainnya.



Gambar 2.4. *Hub and spoke*

(Designing Interfaces, 2010)

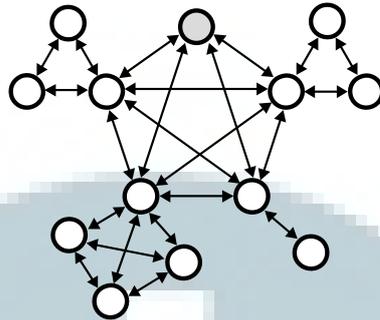
2. *Fully connected*: semua halaman yang tersedia saling terhubung satu dengan yang lain yang memungkinkan pengguna untuk berpindah ke halaman mana pun dalam satu klik.



Gambar 2.5. *Fully connected*

(Designing Interfaces, 2010)

3. *Multi-level*: halaman-halaman yang terhubung satu dengan yang lain hanya halaman utama, sedangkan sublamn hanya terhubung diantara sublamn lainnya.



Gambar 2.6. *Multi-level*

(Designing Interfaces, 2010)

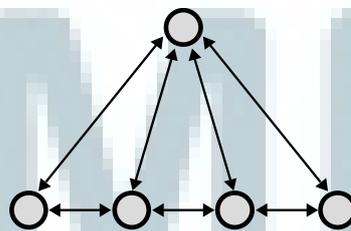
4. *Stepwise*: pengguna hanya dapat menelusuri halaman dalam urutan yang ditentukan dan disediakan *Back/Next link* untuk kembali ke halaman sebelumnya.



Gambar 2.7. *Stepwise*

(Designing Interfaces, 2010)

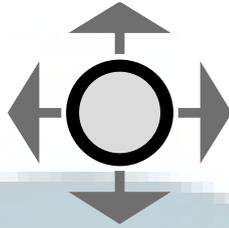
5. *Pyramid*: memulai dari *hub*, pengguna dapat menelusuri subhalaman lainnya secara berurutan dari satu subhalaman ke subhalaman lainnya. Pengguna disediakan dengan *Back/Next link* untuk kembali ke halaman sebelumnya dan dapat kembali ke *hub* kapan pun dan dari subhalaman mana pun.



Gambar 2.8. *Pyramid*

(Designing Interfaces, 2010)

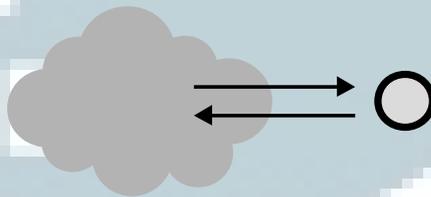
6. *Pan-and-zoom*: halaman direpresentasikan dalam bentuk halaman tunggal yang dapat dikendalikan dengan *pan* maupun *zoom*.



Gambar 2.9. *Pan-and-zoom*

(Designing Interfaces, 2010)

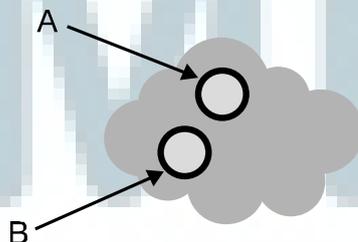
7. *Flat navigation*: beberapa aplikasi tidak memerlukan sistem navigasi, melainkan hanya *tools* dan *functions* yang telah disediakan dalam bentuk *menu*, *toolbar* dan *palette*.
8. *Modal panel*: halaman *modal panel* umumnya digunakan untuk menyampaikan pesan dan tidak ditampilkan dengan pilihan navigasi. Pada umumnya *modal panel* muncul secara menumpuk di atas layar *full screen*.



Gambar 2.10. *Modal panel*

(Designing Interfaces, 2010)

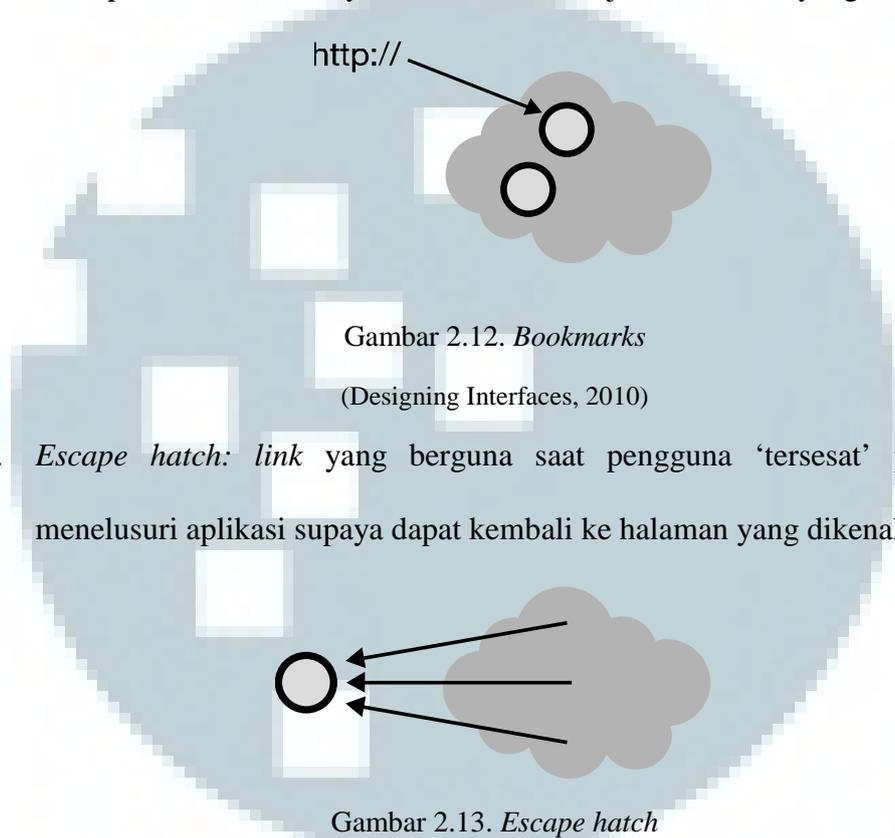
9. *Clear entry points*: digunakan untuk menuntun pengguna bagaimana harus memulai menelusur dalam sebuah aplikasi yang rumit.



Gambar 2.11. *Clear entry points*

(Designing Interfaces, 2010)

10. *Bookmarks*: berperan seperti jalan pintas dalam menuju halaman tujuan yang ingin dikunjungi oleh pengguna kapan saja, berguna supaya pengguna tidak perlu melintas banyak *link* untuk menuju ke halaman yang diinginkan.



Gambar 2.12. *Bookmarks*

(Designing Interfaces, 2010)

11. *Escape hatch*: *link* yang berguna saat pengguna ‘tersesat’ pada saat menelusuri aplikasi supaya dapat kembali ke halaman yang dikenal.

Gambar 2.13. *Escape hatch*

(Designing Interfaces, 2010)

Sebuah desain navigasi yang bagus adalah sistem navigasi yang dapat menjelaskan konten dan struktur interaktif, meningkatkan kegunaan dokumen dan mengakomodasi keperluan pengguna. Dalam membuat skema navigasi yang bagus dan mudah dimengerti, diperlukan *graphic user interface* yang didesain dengan baik. Maka dari itu, berdasarkan fitur yang tersedia dalam aplikasi mengenai pertolongan pertama, *navigational model* yang akan digunakan adalah model *multilevel*.

2.3.2. *Wireframe*

Setiap aplikasi yang berhasil tentunya berawal dari struktur *wireframe* yang teratur. Benyon (2007) memaparkan bahwa *wireframe* merupakan kerangka dasar dari tata letak halaman. Berbagai komponen halaman dirakit menjadi sebuah struktur tertentu menggunakan *wireframe* (hlm. 386). Proses *wireframing* berguna untuk memvisualisasikan alur dan struktur sebuah aplikasi. Menurut Paul Boag (seperti yang dikutip Bank & Cao, n.d.), proses *wireframing* mengirit waktu dalam menguji coba aplikasi yang dirancang sejak awal dengan keuntungan mendapat *feedback* dari pengguna mengenai fitur yang disukai, hirarki visual yang sesuai, dan berbagai macam elemen lainnya tanpa harus mengeluarkan biaya yang berlebihan.

2.3.3. *Graphic User Interface*

Galitz (2007) menjelaskan bahwa *user interface* merupakan sebuah sistem pada komputer yang dapat dimengerti dan diarahkan oleh pengguna. *User interface* berperan sebagai sarana pemberian perintah. Dalam *graphical interface*, pengguna berinteraksi dengan sekumpulan elemen yang disebut '*objects*'. *Objects* tersebut digunakan untuk menjalankan serangkaian tugas tertentu sesuai perintah pengguna (hlm. 16).

Lazar, Feng, & Hochheiser (2010) menjelaskan tentang *eight golden rules of interface design* oleh Shneiderman. Delapan aturan perancangan *user interface* tersebut adalah *strive for consistency*, *cater to universal usability*, *offer informative feedback*, *design dialogs to yield closure*, *prevent errors*, *permit easy*

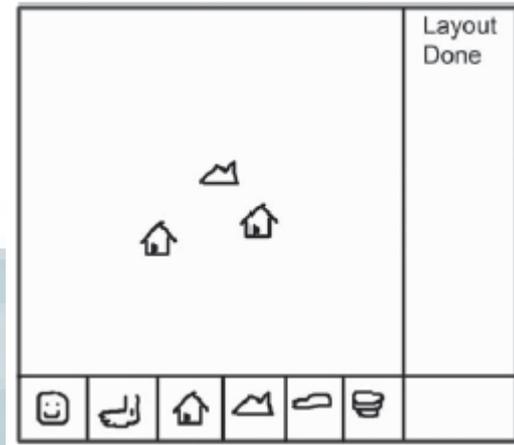
reversal of actions, support internal locus of control dan *reduce short-term memory load* (hlm. 257).

Demi menghasilkan aplikasi yang berguna, *graphic user interface* dari aplikasi tersebut harus menuruti prinsip-prinsip yang telah dijelaskan. Aplikasi yang dirancang disajikan dengan tampilan sederhana namun menarik secara estetika dengan elemen visual yang konsisten sehingga dapat meminimalisir *error* untuk mempermudah pengguna pada saat menelusuri aplikasi tersebut.

2.3.4. Layout

Dibalik sebuah karya yang berhasil, terdapat struktur visual yang diorganisir dengan baik. Penataan elemen-elemen yang terkandung dalam karya tersebut disusun sedemikian rupa sehingga dapat menyampaikan pesan dengan efektif. Graham (1999) menjelaskan bahwa *layout* adalah sebuah bagian dari *interface design* yang fokus pada membuat tampilan pada layar terlihat seimbang, estetis dan mudah dibaca. *Layout* melibatkan penyusunan elemen-elemen sebuah *interface* seperti gambar, media dan *interactive controls* dengan teratur supaya nyaman dipandang (hlm. 90).

Menurut Olsen (2010), fitur yang paling menonjol pada sebuah desain interaktif adalah *layout*. Desain interaktif sangat terpengaruh oleh mekanisme manajemen *layout*. Desain interaktif yang pertama kali diterbitkan adalah *Buxton's MenuLay*. Sketsa objek digambar menggunakan pen dan kemudian digabung lalu diatur dalam palet *layout*.



Gambar 2.14. Struktur awal *layout* layar

(Building Interactive Systems: Principles for Human-Computer Interaction, 2010)

Sistem tersebut digunakan karena sangat praktis dibandingkan dengan harus mengubah angka integer dalam *source code*, menggabung ulang komponen-komponen baru kemudian melihat hasilnya. Sistem tersebut hanya memerlukan desainer untuk meletakkan objek satu per satu ke dalam posisi yang terlihat pas (hlm. 188).

Bagi seorang desainer, sistem *grid* menjadi salah satu alat yang paling berguna dalam menata letak. Dalam menata letak, sistem *grid* membantu dalam membuat struktur *layout* yang konsisten serta membantu dalam menata elemen yang penting supaya terlihat lebih dahulu dari elemen-elemen lainnya (Cooper et al., 2014, hlm. 415).

Dalam menyusun tata letak, kunci untuk menghasilkan *layout* yang berhasil adalah tampilan konsisten dan rapi. Perancangan aplikasi pertolongan pertama akan dilengkapi dengan dasar *layout* menggunakan sistem *grid* supaya pengguna dapat membedakan elemen mana yang menjadi *emphasis*.

2.3.5. Tipografi

Tipografi berperan penting dalam *user interface* karena sebagian besar informasi yang ingin disampaikan disediakan dalam bentuk teks. Apabila pemilihan penggunaan tipografi dalam sebuah *user interface* tidak benar, pembaca akan mengalami kesulitan untuk menangkap pesan yang ingin disampaikan.

Menurut Graham (1999), tipografi adalah seni mendesain *type* yang dapat dibaca, ekspresif dan menarik. Dalam merancang tipografi yang bagus berarti melibatkan pertimbangan pemilihan *typeface*, ukuran, ketebalan huruf, jarak, warna dan layout yang tepat supaya pesan dapat tersampaikan dengan baik. *Typeface* terbagi menjadi beberapa kategori sebagai berikut (hlm. 115-118).

1. *Roman*

Karakteristik dari *typeface* kategori ini adalah tampilan huruf-huruf yang memiliki garis-garis kecil pada ujung hurufnya. Garis-garis kecil ini disebut *serif*. *Roman typeface* pada umumnya memiliki variasi ketebalan yang bermacam-macam pada tubuh huruf.

2. *Sans Serif*

Arti dari kata “*sans serif*” sendiri berarti “*without serifs*”. Kategori *typeface* ini tidak memiliki garis-garis kecil seperti yang terdapat pada *Roman typeface*. *Sans serif typeface* memberikan penampilan yang lebih modern.

3. *Square serif*

Perbedaan *square serif* dari *roman typeface* terdapat pada ujung serif yang memiliki ujung lebih tebal.

4. *Script*

Script typeface memberikan kesan tulisan yang ditulis tangan. Beberapa *script typeface* sulit dibaca, khususnya jika dalam ukuran yang kecil atau semua huruf yang ditulis adalah huruf besar.

5. *Decorative*

Decorative typeface tidak memiliki karakteristik tertentu dan pada umumnya tidak digunakan untuk penulisan *body text* karena sulit dibaca.

Typeface ini cenderung digunakan untuk penulisan *headline*.

Dalam proses pemilihan *typeface* yang tepat untuk proyek yang sedang dikerjakan, desain perlu menguji keterbacaannya dengan cara melihatnya langsung pada layar. *Typeface* yang dapat dibaca pada dokumen yang dicetak tidak tentu selalu dapat dibaca pada saat ditampilkan pada layar. Karakteristik *typeface* yang kecil menjadi sangat sulit dibaca saat ditampilkan di layar karena disebabkan oleh resolusi yang rendah. Aturan yang sama berlaku dari segi warna dan kontras untuk *typeface* yang ditampilkan pada layar dan yang dicetak. Warna yang kontras antara *background* dan *typeface* membuat teks yang ditampilkan mudah dibaca.

Tidwell (2010) menerangkan bahwa dalam memilih *font* yang akan digunakan untuk ditampilkan pada layar, terdapat beberapa poin yang perlu diingat seperti (hlm. 490):

1. *Font* jenis *sans-serif* lebih cocok untuk digunakan pada layar komputer,
2. Hindari penggunaan *italic*, *cursive* dan *ornamental fonts* karena tidak akan terbaca dalam ukuran kecil,
3. *Font* yang berbentuk sangat geometris sulit untuk dibaca karena huruf-huruf yang memiliki bentuk serupa susah dibedakan,
4. Penggunaan *all-caps* untuk *body text* sulit untuk dibaca dan sebaiknya hanya digunakan untuk *headline* atau teks yang singkat,
5. Susun teks dalam *column* dengan kelebaran sedang apabila terdapat teks yang banyak.

Setelah mengetahui lebih lanjut mengenai pemilihan *typeface* untuk layar, jenis *font* yang akan digunakan dalam aplikasi pertolongan pertama adalah *sans-serif* supaya pengguna tidak mengalami kesulitan dalam membaca. Penyediaan informasi dalam bentuk teks juga dibuat singkat supaya menjadi jelas dan mudah untuk dibaca.

2.3.6. Ilustrasi

Ilustrasi pada dasarnya digunakan untuk membantu para pengamat memvisualisasi suatu gambaran. Menurut Zeegen (2009), ilustrasi pada umumnya dianggap sebagai sebuah *graphic art*. Selain disebut sebagai *graphic art*, ilustrasi juga disebut sebagai *commercial art*. Hal ini disebabkan oleh banyaknya karya

ilustrasi yang dibuat bukan sebagai ekspresi diri, melainkan untuk sebuah tugas sesuai permintaan klien.

Ilustrasi merupakan satu-satunya cara untuk berkomunikasi sebelum bahasa tertulis diciptakan. Kehadiran ilustrasi pada masa kini membantu manusia untuk mencatat, mendeskripsi dan mengkomunikasikan seluk-beluk kehidupan.

Sampai saat ini, ilustrasi hadir sebagai salah satu bentuk komunikasi visual yang sangat menjurus. Ilustrasi terus berkembang melampaui batas-batas serta tradisi yang ada sekarang ini (hlm. 6). Beberapa ilustrasi yang umum digunakan pada aplikasi digital adalah:

1. *Vector Graphics*

Vector Graphics memiliki peran yang besar dalam dunia ilustrasi dan desain kontemporer. Ilustrasi bergaya vektor merupakan salah satu gaya ilustrasi yang dapat dibuat dengan menggunakan *vector-editing programs*, salah satu contohnya adalah Adobe Illustrator. Berbeda dengan *bitmap*, vektor tidak tergantung pada resolusi sehingga kualitas gambar akan tetap sempurna pada saat diperbesar. Penggunaan vektor menjadi sangat bermanfaat dalam pembuatan logo, ikon, ilustrasi dan sebagainya, berkisar dari ukuran sekecil thumbnail untuk perangkat *mobile* berlayar kecil sehingga ukuran sebesar *billboard* (Harris & Withrow, 2008, hlm. 10-13).

2. Ikon

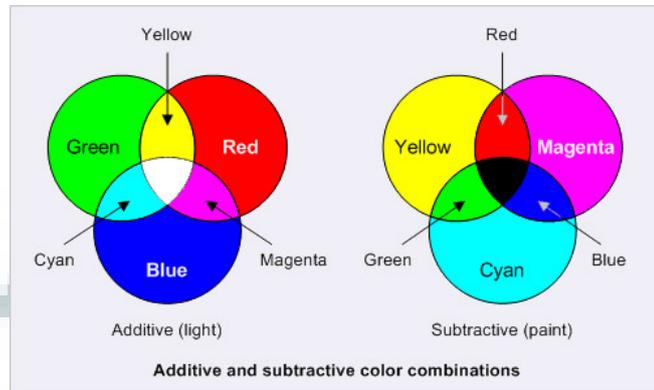
Galitz (2007) menjelaskan bahwa ikon merupakan *pictorial images* yang digunakan untuk mewakili objek dan tindakan (hlm. 652) Dalam memilih

ikon, sebaiknya gunakan ikon yang sudah ada karena akan lebih familiar. Jika ingin membuat ikon yang baru, disarankan untuk menggunakan ikon yang merepresentasikan kata benda, bukan kata kerja. Ikon yang dibuat juga diusahakan sesederhana mungkin dan konsisten supaya tidak membingungkan pengguna (hlm. 659-660). Kegunaan icon maupun simbol dalam ilustrasi sudah menjadi sangat populer. Icon dan simbol menjadi penting dan sangat diperlukan karena kurangnya bahasa umum yang dapat dimengerti oleh semua orang yang menjadi penghalang untuk berkomunikasi antara satu dengan yang lain (Zeegen, 2009, hlm. 90).

Penggunaan ilustrasi yang akan disediakan dalam aplikasi meliputi ikon dan ilustrasi vektor yang berperan sebagai gambar panduan melakukan tindakan pertolongan pertama. Sebagian besar ikon yang akan digunakan dipilih dari ikon yang sudah ada sebelumnya untuk menjaga familiaritas dengan pengguna dan menghindari salah pengertian.

2.3.7. Warna

Dalam desain, warna merupakan salah satu elemen yang penting. Salah satu kegunaan warna adalah untuk membangun suasana dan memberikan kesan. Warna terbagi menjadi dua jenis yaitu RGB (*red, green, blue*) yang merupakan *additive colour* dan CMYK (*cyan, magenta, yellow, black*) yang merupakan *subtractive colour*.



Gambar 2.15. Warna RGB dan CMYK

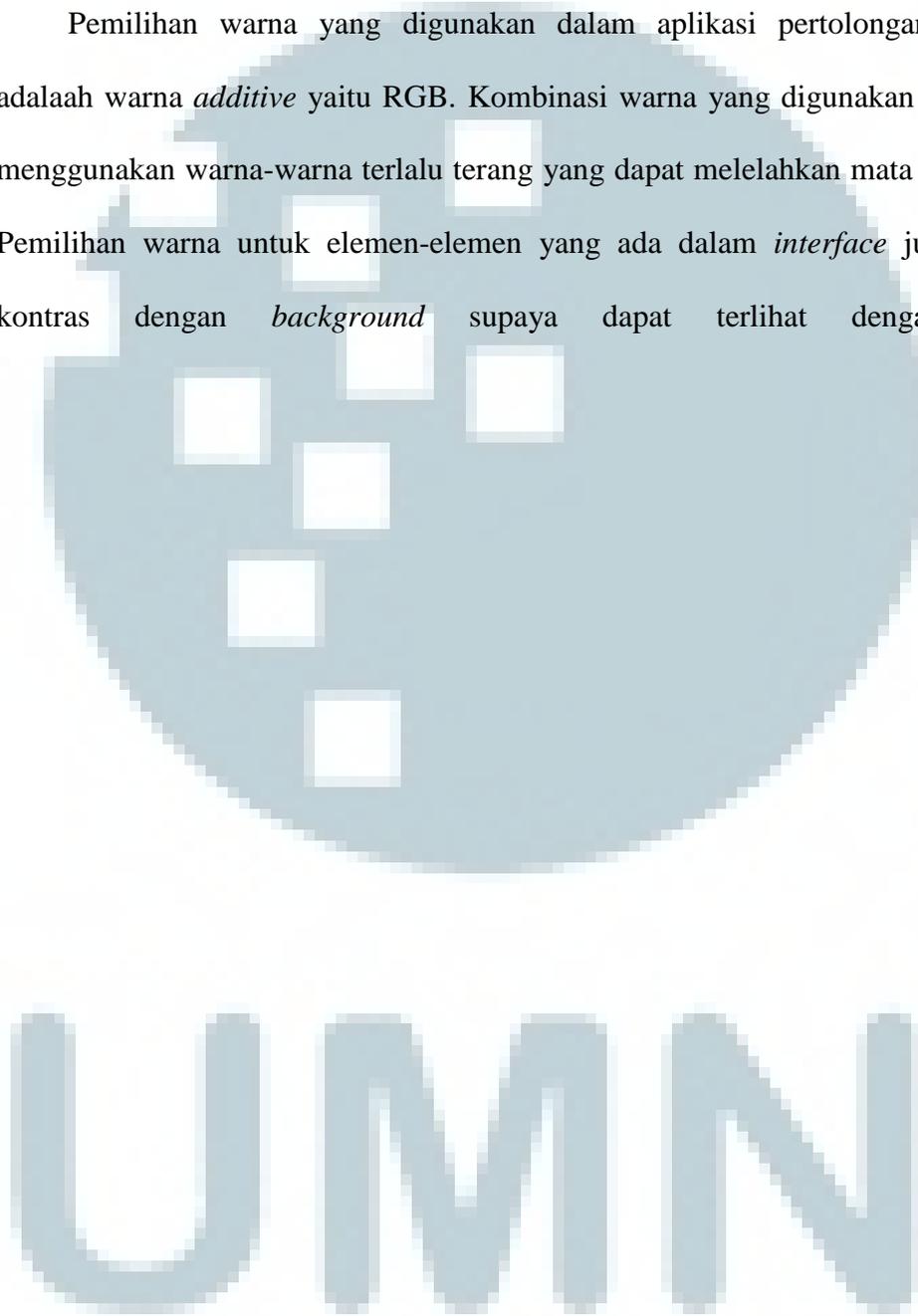
(<https://intranet.mcad.edu/kb/should-i-work-rgb-or-cmyk-color-management-mcad>, n.d.)

Menurut Olsen (2010), semua komputer dan televisi menggunakan sistem warna RGB atau dengan nama lain *red, green, blue colour system*. Kepadatan pixel RGB pada umumnya disebut sebagai *spatial resolution*. Satuan *spatial resolution* dinyatakan sebagai *dots per inch* (DPI) atau *dots per centimeter* (hlm. 21).

Graham (1999) mengatakan bahwa warna berperan penting dalam pembuatan sebuah dokumen interaktif. Warna mampu membangun respon emosional dan juga meningkatkan atau mengurangi keterbacaan sebuah dokumen jika tidak digunakan dengan tepat (hlm. 102). Dalam pemilihan penggunaan warna, terdapat beberapa aturan penggunaan sebagai panduan seperti kombinasi warna *warm* dan *cool* yang sebenarnya dapat menjadi efektif untuk menghasilkan tampilan yang *balanced*, penggunaan warna *background* yang terang lebih tipikal dengan *interface* komputer, kontras tinggi yang menimbulkan penekanan sedangkan kontras rendah terasa lebih menenangkan, *saturated colours* yang dapat melelahkan mata jika digunakan secara berlebihan dan terakhir berbagai

kombinasi warna yang memberikan beragam kesan berbeda (Tidwell, 2010, hlm. 489).

Pemilihan warna yang digunakan dalam aplikasi pertolongan pertama adalah warna *additive* yaitu RGB. Kombinasi warna yang digunakan juga tidak menggunakan warna-warna terlalu terang yang dapat melelahkan mata pengguna. Pemilihan warna untuk elemen-elemen yang ada dalam *interface* juga dibuat kontras dengan *background* supaya dapat terlihat dengan jelas.



UMN