



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk mengubah, 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 Desain

Menurut Landa (2014), grafik desain merupakan bentuk komunikasi visual yang bertujuan untuk memberikan informasi kepada audience. Grafik desain bukan hanya karya melainkan solusi yang dapat memberi informasi, mengajak, memberikan ide, memotivasi, memberikan identity, dan membawa berbagai pesan (Landa, 2014, hlm. 1).

##### 2.1.1. Formal Elements

*Formal elements* merupakan *basic* dari grafik desain. Setiap elemen mempunyai keunikan dan potensinya masing-masing. Dengan mengetahui potensi dari setiap elemen, desainer dapat memanfaatkannya untuk merancang desain yang komunikatif dan juga ekspresif (Landa, 2014, hlm. 19). Berikut *formal elements* untuk 2D desain :

##### 2.1.1.1. Garis / Line

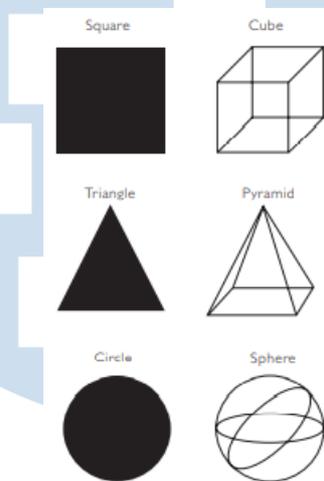
Garis merupakan titik yang panjang dan terarah. Garis dapat dibuat oleh berbagai alat seperti pensil sampai dengan *cotton swab* yang diberi tinta. Peran garis sangat besar dalam memberikan komunikasi dan membuat komposisi yang baik. Saat menggambar garis, garis tersebut akan memberikan informasi. Informasi tersebut dapat didapatkan dalam bentuk garis yang lurus, belok, ataupun yang lain. Oleh karena itu, garis memimpin mata yang melihat dengan berbagai makna seperti garis yang lembut atau keras, putus-putus atau mulus, tebal atau tipis, dan seterusnya (Landa, 2014, hlm. 19).



Gambar 2.1. Garis  
Sumber: Landa (2014)

### 2.1.1.2. Bentuk / *Shape*

Berbeda dengan garis yang bersifat terbuka, bentuk bersifat tertutup. Bentuk sendiri merupakan garis yang saling menyatu dan membuat bentuk yang tertutup setiap arahannya. Ada 3 bentuk dasar yaitu kotak, segitiga, dan lingkaran. Dari ketiga bentuk dasar ini dapat membentuk kualitas desain yang berbeda-beda, baik dari 2 dimensi hingga 3 dimensi (Landa, 2014, hlm. 20-21).



Gambar 2.2. Bentuk  
Sumber: Landa (2014)

### 2.1.1.3. Tekstur / *Texture*

Tekstur dapat dibagi menjadi 2 kategori yaitu *visual* dan *tactile*. *Tactile* merupakan tekstur yang dapat diraba dan dirasakan secara fisik. Sedangkan *visual* hanya merupakan ilusi yang dibuat oleh tangan manusia. Misalnya tekstur yang dibuat dari foto, gambar, *pattern*, ataupun ilustrasi (Landa, 2014, hlm. 28).



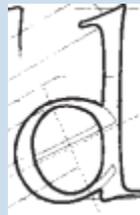
Gambar 2.3. Tekstur *Tactile* dan *Visual*  
Sumber: Landa (2014)

### 2.1.2. Tipografi

Landa (2014) menyatakan bahwa *typeface* merupakan gabungan karakter yang konsisten dengan karakteristik yang unik. Biasanya pada karakter *typeface* meliputi simbol, angka, berbagai tanda, dll (hlm. 44). Menurut Landa (2014), ada beberapa tipe klasifikasi *typeface* yaitu (hlm. 47-48):

#### 2.1.2.1. *Old style*

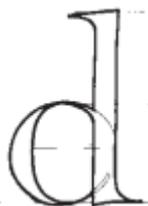
Gaya *typeface* yang lama dengan kesan romawi. *Old style* pertama kali dipublikasi pada abad kelima belas. Karakteristik khas *typeface* ini adalah menggunakan *angled* dan *bracketed serif*. Sehingga *typeface* ini dapat dikenal dengan melihat bracketnya yang diagonal.



Gambar 2.4. *Old style*  
Sumber: Landa (2014)

#### 2.1.2.2. *Transitional*

*Typeface transitional* berasal dari abad kedelapan belas yang melambangkan pergantian zaman dari *old* ke *modern*. Maka dari itu, *transitional* memiliki desain gabungan dari *old style* dan *modern*.

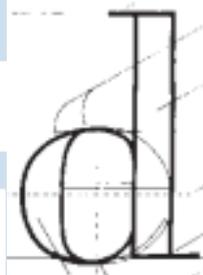


Gambar 2.5. *Transitional*  
Sumber: Landa (2014)

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

### 2.1.2.3. *Modern*

*Typeface modern* terbentuk pada akhir abad kedelapan belas dan awal abad kesembilan belas. *Typeface* ini cenderung lebih geometris dan tegas dibandingkan *old style* dan *transitional*. Serta merupakan *typeface* romawi yang paling simetris.



Gambar 2.6. *Modern*  
Sumber: Landa (2014)

### 2.1.2.4. *Slab Serif*

*Slab serif* diperkenalkan pada awal abad kesembilan belas dengan gaya yang tebal dan berat. Salah satu contoh *slab serif* adalah *Memphis*, *Clarendon*, dan *Bookman*.



Gambar 2.7. *Slab Serif*  
Sumber: Landa (2014)

### 2.1.2.5. *Sans Serif*

*Typeface sans serif* berasal dari awal abad kesembilan belas dengan ciri khas unik yaitu tanpa menggunakan *serif*. Salah satu contohnya adalah *Futura*, *Helvetica*, dan *Univers*.



Gambar 2.8. *Sans Serif*  
Sumber: Landa (2014)

### 2.1.2.6. *Blackletter*

Berawal dari abad ketiga belas hingga abad kelima belas, *blackletter* terkenal unik dan *gothic*. *Typeface* ini juga mempunyai karakteristik berupa *stroke* yang tebal dan huruf-huruf yang tebal namun elok karena *condensed*.



Gambar 2.9. *Blackletter*  
Sumber: Landa (2014)

### 2.1.2.7. *Script*

*Script* mempunyai karakter yang mirip dengan tulisan tangan. *Typeface* ini cenderung mempunyai karakter yang miring dan nyambung seperti tulisan halus kasar. Dikarenakan karakter yang menyerupai halus kasar, *typeface* ini seperti ditulis menggunakan pena, pensil, maupun kuas.



Gambar 2.10. *Script*  
Sumber: Landa (2014)

### 2.1.2.8. *Display*

*Typeface display* cenderung mewah, besar dan dekoratif. Maka dari itu, *typeface* ini cocok untuk digunakan pada judul dan tidak untuk *text type*. Hal ini dikarenakan *typeface display* yang dekoratif akan sulit dibaca bila diletakkan pada *text type*.



Gambar 2.11. *Display*  
Sumber: Landa (2014)

### 2.1.3. Warna

Berdasarkan pendapat Landa (2014), pandangan akan warna berbeda pada setiap orang. Pandangan tersebut bergantung pada pengalaman diri masing-masing. Pengalaman tersebut dapat berupa budaya, agama, jenis kelamin, maupun gaya hidup pribadi (hlm. 129).

#### 2.1.3.1. Warna Primer

Tiga warna utama atau primer yaitu merah, hijau, dan biru (RGB). Ketiga warna ini merupakan yang utama karena bila warna tersebut saling dicampurkan akan membuat warna baru. Misalnya bila ketiga warna RGB digabungkan akan membentuk warna putih. Namun, bila warna lain dicampurkan tidak akan menghasilkan ketiga warna utama ini, Oleh karena itu, merah, hijau, dan biru menjadi warna primer. (Landa, 2014, hlm. 24).



Gambar 2.12. Warna Primer  
Sumber: Landa (2014)

#### 2.1.3.2. Elemen Warna

Menurut Landa (2014), ada beberapa elemen warna yaitu *hue*, *value*, dan juga *saturation* (hlm. 23):

##### 1. *Hue*

*Hue* merupakan nama atas warna seperti hijau atau merah dan biru atau oranye. warna *hue* dapat dibedakan menjadi dua temperatur warna

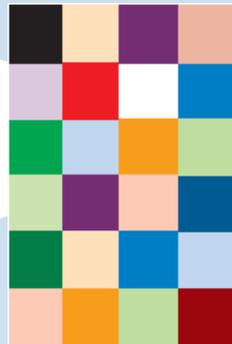
yaitu *cool* dan *warm*. Kedua temperatur ini berperan besar dalam menyampaikan ekspresi dan perasaan (hlm. 129-131).



Gambar 2.13. *Hue* dan Temperatur warna  
Sumber: Landa (2014)

## 2. *Value*

*Value* mengarah ke tingkat terang ataupun gelapnya suatu warna. Dalam mengatur *value* suatu warna dibutuhkan campuran 2 warna netral. Dua warna tersebut adalah warna hitam dan putih murni. Hitam dan putih merupakan warna (*pigment*) tetapi tidak dianggap *hue* karena tidak ditemukan spektrum yang terlihat pada kedua warna tersebut. Oleh karena itu, warna hitam dan putih merupakan warna netral atau *achromatic*.



Gambar 2.14. *Value Contrast*  
Sumber: Landa (2014)

## 3. *Saturation*

*Saturation* lebih mengarah ke tingkat kecerahan atau kekusaman dari suatu warna. *Hue* yang mencapai tingkat kecerahan maksimal tidak memiliki campuran warna netral (putih atau hitam) ataupun campuran warna netral (abu-abu). Sedangkan *hue* yang dicampur dengan warna netral/campuran warna netral akan menghasilkan *hue* yang kusam dengan berbagai tingkatan.



Gambar 2.15. *Saturation*  
Sumber: Landa (2014)

### 2.1.3.3. Psikologi Warna

Samara (2007) menyatakan bahwa setiap warna memiliki makna dan arti yang berbeda. Dengan memanfaatkan makna dari warna, desainer dapat memberikan karya yang emosional dari warna saja (hlm. 110-111). Berikut beberapa psikologi warna dari Samara (2007, hlm. 110-111):

#### 1. Merah

Warna merah memberi kesan semangat dan gairah yang membara. Selain itu, warna merah juga dapat memberikan kesan menyeramkan dan menakutkan. Merah seolah meninggalkan pesan “lawan atau lari”.



Gambar 2.16. Merah  
Sumber: Samara (2007)

#### 2. Biru

Biru merupakan warna yang paling mudah disukai dibanding warna lain. Hal ini karena biru yang memberikan kesan tenang dan juga melindungi. Warna biru yang seperti laut dan langit juga memberikan persepsi yang kuat serta dapat dipercaya.



Gambar 2.17. Biru  
Sumber: Samara (2007)

### 3. Kuning

Matahari dan kehangatan menjadi lambang dari warna kuning. Warna kuning memiliki makna kebahagiaan karena setiap warna yang dicampur dengan warna kuning akan menjadi cerah. Kuning juga berarti pikiran yang bersih dan ingatan yang baik.



Gambar 2.18. Kuning  
Sumber: Samara (2007)

### 4. Coklat

Warna coklat lekat dengan impresi akan bumi dan kayu sehingga memberikan kesan yang ramah, nyaman, dan aman. Warnanya yang lekat dengan bumi juga terlihat terpercaya dan tahan lama. Coklat yang natural juga dipandang memiliki sifat yang kasar, pekerja keras, dan ekologis.



Gambar 2.19. Coklat  
Sumber: Samara (2007)

### 5. Violet

Violet memberikan pandangan yang sulit dipercaya, misterius, dan sukar dipahami. *Value* dan *hue* dari warna violet sangat berpengaruh terhadap makna dari warna tersebut. Misalnya seperti warna violet dengan *value* rendah melambangkan kematian.



Gambar 2.20. Violet  
Sumber: Samara (2007)

## 6. Hijau

Alam dan vegetarian melekat erat dengan warna hijau. Dikarenakan 2 hal itu, warna hijau menjadi warna yang paling menyenangkan dan nyaman dibanding warna yang lain. Warna hijau yang terang juga memberi kesan awet muda. Akan tetapi warna hijau yang memiliki *saturation* rendah dapat berarti penyakit atau kerusakan.



Gambar 2.21. Hijau  
Sumber: Samara (2007)

## 7. Oranye

Warna oranye merupakan gabungan dari warna merah dan kuning. Sehingga arti dari warna merah dan kuning juga dapat masuk ke warna oranye. Misalnya gairah pada warna merah dan kehangatan pada warna kuning. Warna oranye sendiri melambangkan sifat yang *outgoing* dan penuh akan rasa bertualang. Namun, warna oranye juga dapat dipandang kurang bertanggung



jawab.

Gambar 2.22. Oranye  
Sumber: Samara (2007)

## 8. Abu-abu

Abu-abu merupakan warna paling netral dari warna yang lain. Warna ini dapat dipandang sebagai keraguan tetapi juga dapat terpancang bijaksana dan berwibawa. Tergantung dari cara pemakaiannya, warna abu-abu juga dapat berkaitan dengan teknologi, kompetensi, kemewahan, kontrol, dan juga presisi.



Gambar 2.23. Abu-abu  
Sumber: Samara (2007)

### 2.1.3.4. Color Schemes

Menurut Landa (2014), dalam mendesain dengan warna harus selalu memperhatikan *hue*, *value*, dan *saturation*. Ketika mengubah *value* / *saturation* akan mempengaruhi makna dan cara kerja dari warna/*hue*. *Color schemes* merupakan penggabungan harmonis dari warna dengan *hue*, *saturation* penuh, dan pertengahan *value*. Berikut beberapa jenis *color schemes* dari *color wheel* (hlm. 132-134):



Gambar 2.24. Color Wheel  
Sumber: Landa (2014)

## 1. *Monochromatic*

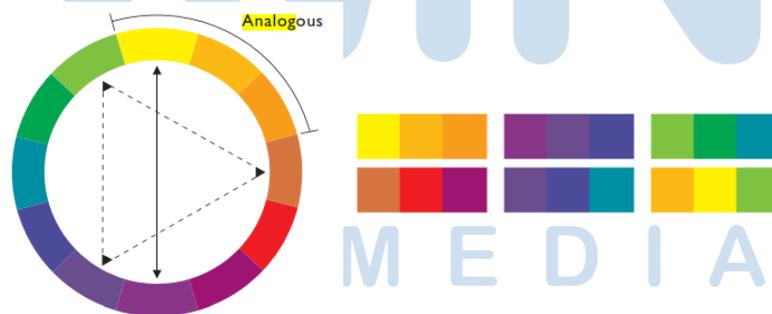
*Color schemes monocromatic* hanya terdiri dari 1 *hue* yang dominan dengan permainan warna kontras dari *value* dan *saturation*. *Monochromatic* memberikan visual yang terlihat seimbang dan simple. Skema ini juga dapat menjadi alternatif dari warna hitam untuk proyek yang menggunakan 1 warna saja.



Gambar 2.25. *Monochromatic*  
Sumber: Landa (2014)

## 2. *Analogous*

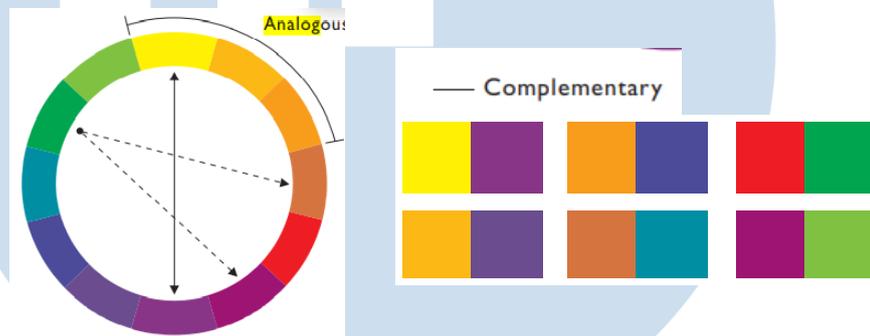
*Analogous* menggunakan tiga warna yang berdekatan dari *color wheel*. Skema *analogous* cenderung harmonis karena warna yang mirip dengan satu sama lain. *Analogous* memberikan kesan kesatuan dan ketenangan. Dalam skema *analogous* ada 1 warna yang mendominasi sedangkan 2 warna lainnya berperan sebagai warna tambahan/*support*.



Gambar 2.26. *Analogous*  
Sumber: Landa (2014)

### 3. *Complementary*

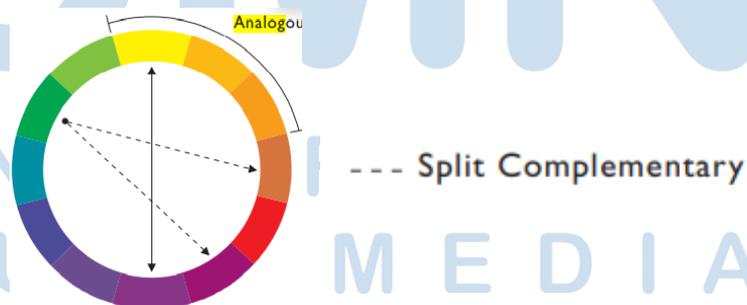
*Complementary* menggunakan 2 warna yang saling bersebrangan dari *color wheel*. Warna yang bersebrangan ini memberikan visual yang menarik karena kontras yang kuat antar kedua warna. Kedua warna *complementary* dapat dicampur untuk menghasilkan abu-abu atau kilauan.



Gambar 2.27. *Complementary*  
Sumber: Landa (2014)

### 4. *Split Complementary*

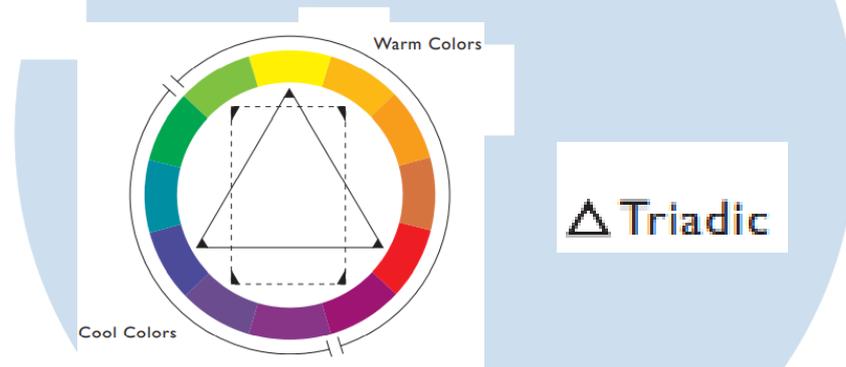
*Split complementary* menggunakan 3 warna yaitu 1 warna dan 2 warna yang berdekatan dengan warna yang bersebrangan. Jadi 2 warna itu adalah kedua warna yang mengelilingi warna yang bersebrangan dengan 1 warna. Warna yang diberikan memiliki visual yang intens tetapi lebih kurang dramatis dibanding *Complementary*.



Gambar 2.28. *Split Complementary*  
Sumber: Landa (2014)

## 5. *Triadic*

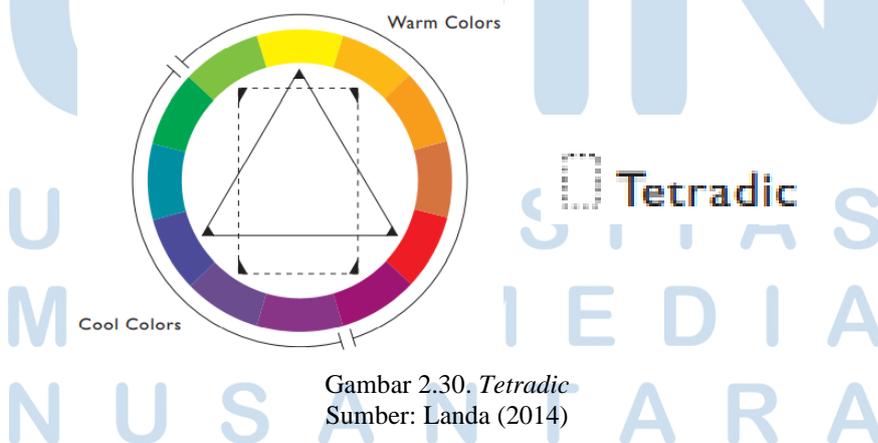
*Triadic* merupakan 3 warna yang memiliki jarak yang sama dari satu sama lain. Misalnya warna red, blue, dan yellow. *Triadic* mempunyai warna dengan keseimbangan yang melekat tetapi memiliki visual yang berbeda dnegan mempertahankan harmoni masing-masing warna.



Gambar 2.29. *Triadic*  
Sumber: Landa (2014)

## 6. *Tetradic*

*Tetradic* terdiri dari 4 warna dalam 2 set *complementary*. Skema ini memberikan perbedaan *hue* dan kontras yang besar. Sehingga untuk desainer pemula akan sulit mengharmonikasi 4 warna ini. Salah satu cara untuk mengharmonisasi keempat warna ini adalah dengan memberikan 1 *hue* dominan dan 3 lainnya menjadi *support*.



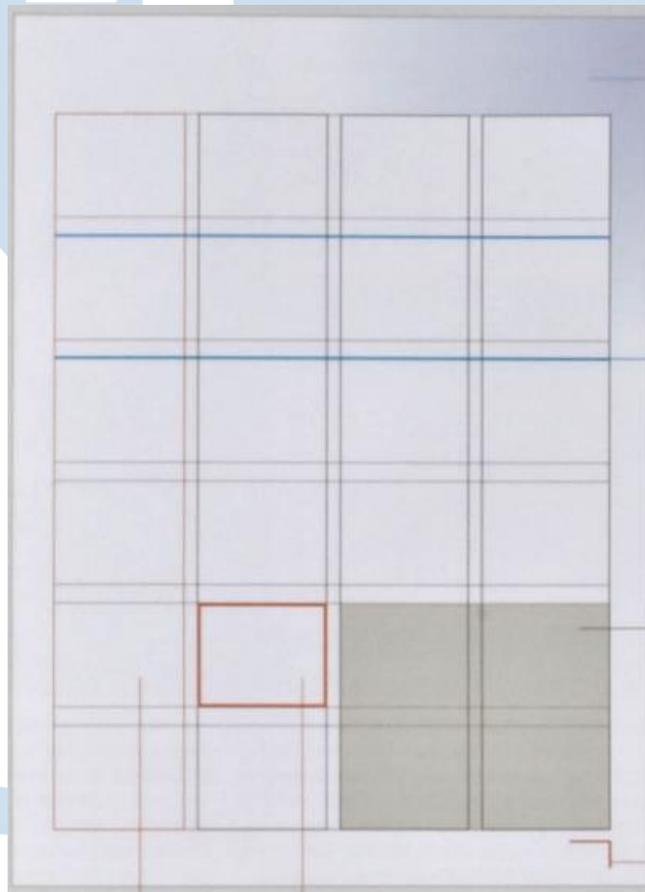
Gambar 2.30. *Tetradic*  
Sumber: Landa (2014)

#### 2.1.4. Grid

Grid adalah struktur yang membentuk suatu buku, *website*, aplikasi, majalah, dan lain-lain. Struktur ini terbentuk dari gabungan garis vertikal dan horizontal yang menciptakan kolom dan margin. Grid membuat tipografi dan gambar menjadi terstruktur dan rapi sehingga pembaca dapat dengan mudah mendapatkan informasi (Landa, 2014, hlm. 175).

##### 2.1.4.1. Anatomi Grid

Menurut Samara (2002), anatomi grid dapat dibagi menjadi 6 yaitu (Hlm. 25):



Gambar 2.31. *Grid*  
Sumber: Samara (2002)

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

### 1. *Columns*

Kolom merupakan garis vertikal sejajar yang berada di antara margin. Kolom terdiri atas satu atau lebih dan mempunyai lebar yang cenderung sama. Namun kolom juga dalam memiliki lebar yang berbeda tergantung dari informasi apa yang ingin disampaikan.



Gambar 2.32. *Columns*  
Sumber: Samara (2002)

### 2. *Modules*

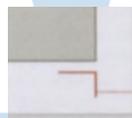
Modul ialah ruang individual yang dan memiliki jarak yang sama. Sehingga ketika ruang individu itu berulang akan membentuk kolom dan baris.



Gambar 2.33. *Module*  
Sumber: Samara (2002)

### 3. *Markers*

*Markers* berada di luar margin dan menjadi tempat indikator untuk teks yang selalu muncul. Misalnya *section titles*, *page*, *folios*, dan segala elemen yang hanya berada di satu lokasi untuk setiap *layout*.



Gambar 2.34. *Markers*  
Sumber: Samara (2002)

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

#### 4. *Spatial zones*

Zona spasial adalah gabungan dari beberapa modul yang menjadi satu. Setiap zona dapat memberikan informasi yang berbeda-beda. Misalnya zona yang panjang secara horizontal cenderung untuk gambar, lalu dibawahnya akan ada zona untuk kolom teks.



Gambar 2.35. *Spatial Zones*  
Sumber: Samara (2002)

#### 5. *FlowLines*

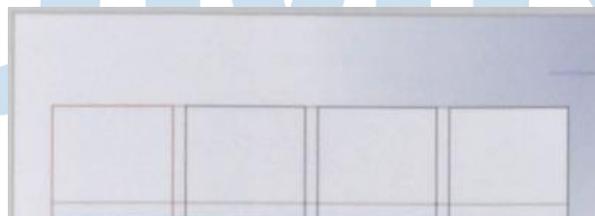
Jajaran garis horizontal yang mematahkan ruangan kosong adalah *flowLines*. *FlowLines* dapat membimbing mata untuk melihat format dan menjadi pemandu untuk awalan serta akhiran gambar atau teks.



Gambar 2.36. *FlowLines*  
Sumber: Samara (2002)

#### 6. *Margins*

Margin adalah *negative spaces* di antara tepi kertas dan isi konten kertas. Sehingga, konten seperti teks dan gambar yang tersusun itu dikelilingi oleh margin. Margin dapat digunakan untuk menjadi pusat perhatian ataupun tempat mata beristirahat dari konten yang banyak. Margin juga berguna untuk menjadi tempat informasi yang tidak terlalu penting.



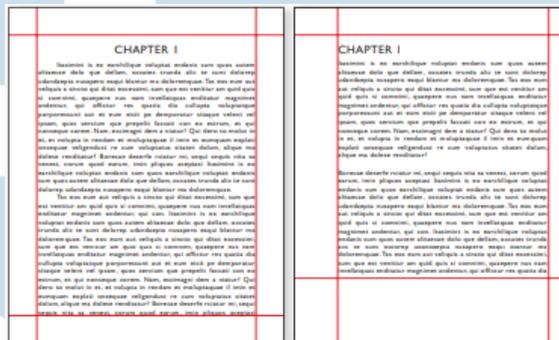
Gambar 2.37. *Margins*  
Sumber: Samara (2002)

### 2.1.4.2. Jenis Grid

Landa (2014) menyatakan bahwa grid dapat terbagi menjadi grid satu kolom, grid multi kolom, dan grid modular (hlm. 175-181).

#### 1. Grid Satu Kolom

Grid yang paling umum dan *basic* dimana hanya ada satu kolom teks yang digunakan pada halaman tersebut. Selain menggunakan satu kolom, grid ini juga memanfaatkan margin pada bagian atas, kiri, kanan, dan bawah kertas. Margin digunakan untuk memastikan pembaca akan tetap mendapatkan konten dengan aman.



Gambar 2.38. *Single-column Grid*  
Sumber: Landa (2014)

#### 2. Grid Multi Kolom

Grid multi kolom terdiri atas dua atau lebih kolom yang berjajar. Kolom yang sejajar memastikan isi yang banyak dan berbeda tetap terstruktur dan tidak keluar jalur. Contohnya seperti tali pada perlombaan renang yang memastikan perenang tidak keluar jalur dan menabrak satu sama lain.



Gambar 2.39. *Multicolumn Grid*  
Sumber: Landa (2014)

### 3. Grid Modular

Grid modular terbentuk dari berbagai modul dan unit-unit individual yang tercipta dari tabrakan antar garis kolom dan flow. Grid ini merupakan grid yang paling fleksibel untuk ilustrasi berat karena gambar dan tulisan dapat diletakkan di satu atau lebih modul.



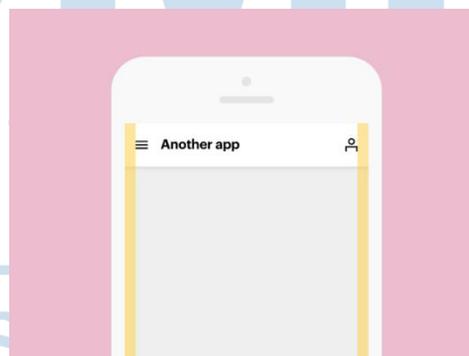
Gambar 2.40. *Modular Grid*  
Sumber: Landa (2014)

#### 2.1.4.2. *Mobile Layouts & Grid*

Reicher dkk (n.d), menyatakan bahwa ada beberapa hal yang tidak dapat dilihat oleh *user*, tetapi dapat dilihat oleh desainer. Desainer dapat meletakkan konten-konten pada tampilan secara mudah dan konsisten dengan menggunakan *layout*. Berikut *layout & grid* dari *mobile apps* :

##### 1. *Margin*

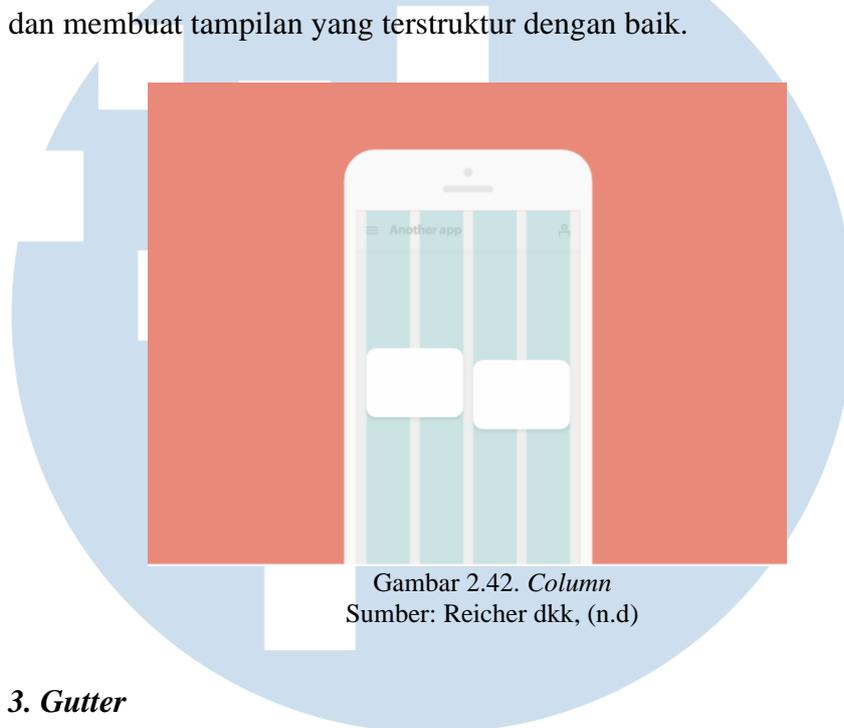
*Margin* adalah jarak diantara konten aplikasi dan bagian ujung layer. *Margin* biasanya memiliki ukuran lebar yang tetap. Biasanya ukuran *margin* berupa 16 px dan terkadang 20 atau 24 px. Ukuran jarak antara elemen UI cenderung kelipatan 8 karena kebanyakan tampilan *mobile* memiliki 8pt *grid*.



Gambar 2.41. *Margin Mobile*  
Sumber: Reicher dkk, (n.d)

## 2. Column

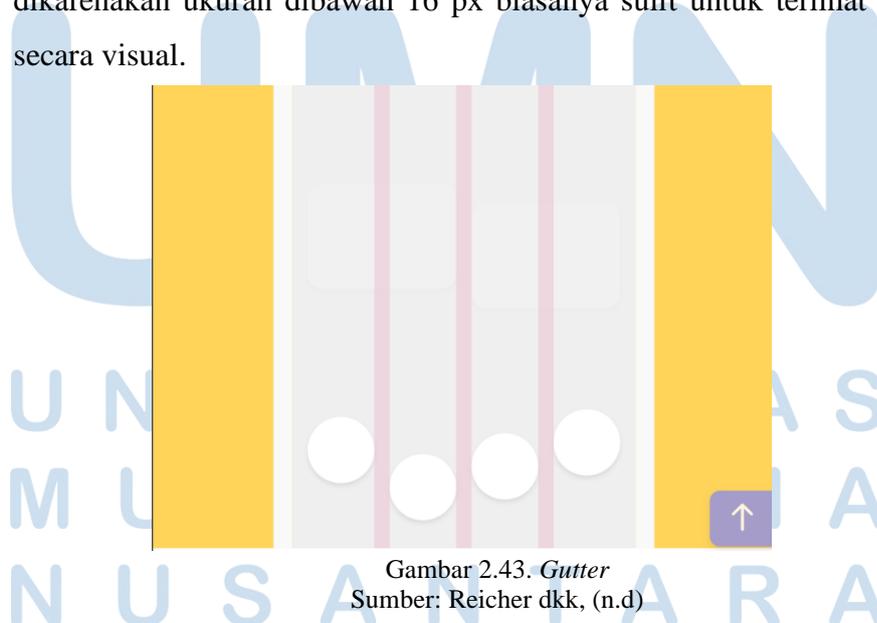
*Column* adalah tempat meletakkan konten-konten di tampilan aplikasi. *Column* membantu desainer untuk menentukan peletakkan elemen dan membuat tampilan yang terstruktur dengan baik.



Gambar 2.42. *Column*  
Sumber: Reicher dkk, (n.d)

## 3. Gutter

*Gutters* berguna untuk memisahkan konten yang ada di dalam *columns*. Ukuran *gutters* yang biasa digunakan adalah 16 px. Hal ini dikarenakan ukuran dibawah 16 px biasanya sulit untuk terlihat terpisah secara visual.



Gambar 2.43. *Gutter*  
Sumber: Reicher dkk, (n.d)

### 2.1.4.3. Grid di *Mobile Apps*

Menurut Zhulidin (2019), grid *mobile apps* ada 4 macam yaitu 2, 4, 8, dan 12 *column grid*. Grid terbaik untuk tampilan aplikasi *mobile* adalah 12 *column grid*. Dengan menggunakan grid ini desainer dapat meletakkan komponen atau elemen yang genap maupun ganjil. Sehingga grid ini akan lebih fleksibel dalam menjaga keseimbangan tampilan. Sedangkan kerugian menggunakan grid 12 kolom adalah 1 kolom yang terlalu kecil sehingga sulit untuk membuat elemen dari 1 kolom saja.



Gambar 2.44. 12 Column Grid  
Sumber: Zhulidin (2019)

## 2.2. *Interaction Design*

*Interaction design* adalah perancangan interaktif untuk membantu *user* dalam berkomunikasi ataupun berinteraksi dalam kehidupan sehari-hari. Hal tersebut berarti *interaction design* adalah tentang membuat *user experience* yang meningkatkan dan menambah pengalaman *user* dalam bekerja, berkomunikasi, dan juga berinteraksi. Terdapat berbagai istilah yang digunakan *interaction design* sesuai aspek yang dirancang yaitu *interface design (UI)*, *software design*, *user-centered design*, *product design*, *web design*, *user experience design*, dan *interactive system design* (Sharp dkk., 2019, hlm. 10).

### **2.2.1. Interaction Types**

Menurut Sharp, dkk. (2019) dari bukunya yang berjudul “*Interaction Design Beyond Human-Computer Interaction*”, terdapat berbagai tipe dari interaksi. Tipe-tipe interaksi tersebut ialah *Instructing*, *conversing*, *manipulating*, *exploring*, dan *responding* (hlm. 81).

#### **2.2.1.1. Instructing**

Interaksi *instructing* adalah tipe yang memberikan *user* untuk melakukan instruksi kepada sistem. Hal tersebut dapat dilakukan dengan berbagai cara seperti memilih *menu*, menuliskan perintah, menekan *button*, serta perintah melalui suara. Keuntungan menggunakan interaksi *instructing* ialah sistemnya yang praktis dan efisien.

#### **2.2.1.2. Conversing**

Interaksi *conversing* merupakan interaksi *user* dengan dialog pada sistem. *User* dapat berinteraksi dengan melalui *interface* atau memilih pertanyaan yang akan dibalas oleh sistem melalui *text* ataupun suara. Tipe *conversing* dirancang untuk menjawab *user* seperti layaknya manusia berkomunikasi. Contoh interaksi ini adalah *chatbots*, *help facilities*, ataupun sistem bantuan. Keuntungan utama dalam menggunakan interaksi *conversing* adalah interaksi yang diberikan membuat *user* menjadi lebih familiar dengan sistem/aplikasi.

#### **2.2.1.3. Manipulating**

Pada interaksi *manipulating*, *user* akan berinteraksi dengan objek pada ruangan fisik ataupun virtual dengan memanipulasi. Memanipulasi dilakukan dengan cara membuka, menutup, menahan, ataupun meletakkan objek. Misalnya, *zoom in/zoom out* terhadap foto yang hanya dapat dilakukan pada objek virtual. Dengan menggunakan interaksi *manipulating* akan mengurangi *anxiety user* dan meningkatkan kepercayaan diri *user* dengan membuat *user* merasa *in control*.

#### **2.2.1.4. Exploring**

Selanjutnya terdapat interaksi *exploring*, *user* akan berpindah melalui lingkungan virtual. Lingkungan virtual mencakupi dunia 3 dimensi (3D) dan *augmented/virtual reality systems (AR/VR)*. Interaksi ini memberikan *user* kebebasan untuk mengeksplor berbagai aspek dari dimensi virtual yang diberikan oleh sistem. Contoh dari interaksi *exploring* adalah pameran virtual kampus, pesta virtual, ataupun *video games* seperti Fortnite.

#### **2.2.1.5. Responding**

Interaksi *responding* adalah interaksi yang diawali oleh sistem lalu memberikan *user* pilihan untuk merespon. Interaksi yang diberikan berupa peringatan, penjelasan ataupun sesuatu yang menunjukkan *interest* dari *user*. Contohnya adalah *fitness tracker* yang memberikan notifikasi kepada *user* apabila *user* telah mencapai *checkpoint* misalnya 10.000 langkah. Sehingga, interaksi *responding* layaknya notifikasi yang dapat direspons ataupun diabaikan oleh *user*.

### **2.3. User Interface**

#### **2.3.1. Teori 8 Golden Rules dari Interface Design**

*User* dapat mengalami kesulitan dalam menavigasikan tampilan aplikasi. Oleh karena itu, desainer dituntut untuk memberikan tampilan dengan aturan yang jelas dan dapat membantu. Dengan aturan yang jelas tidak hanya dapat membantu *user* tetapi juga dapat mengurangi pekerjaan *user* saat membuka aplikasi (Shneiderman & Plaisant, 2005, hlm. 61).

Menurut Shneiderman dan Plaisant (2005), ada 8 prinsip aturan yang dapat diimplementasikan ke sebagian besar sistem interaktif. Delapan prinsip tersebut disebut *golden rules* (hlm. 74).

### **2.3.1.1. Strive For Consistency**

Aturan ini mementingkan konsistensi dalam mendesain tampilan aplikasi. Konsisten dalam merancang dengan pemilihan warna, layout, font, ukuran font, dan *icon* yang serupa atau sama pada setiap tampilan. Namun, ada saatnya aturan ini dilanggar yaitu seperti *warning* pada tampilan. *warning* seperti *password* yang salah tentu akan menggunakan warna yang mencolok dan berbeda dari lainnya (Shneiderman & Plaisant, 2005, hlm.74)

### **2.3.1.2. Cater to Universalizability**

Menurut Shneiderman dan Plaisant (2005), perlu kesadaran akan kepentingan dari rancangan desain yang universal dan dapat digunakan oleh beragam *user*. Pada aturan ini, desainer harus sadar akan adanya perbedaan pengalaman, usia, dan kemampuan berteknologi masing-masing *user*. Oleh karena itu, penting untuk merancang fitur yang ramah terhadap pemula dan juga ahli. Contohnya seperti fitur bantuan untuk *user* pemula dan fitur *shortcut* untuk yang sudah pandai menggunakan teknologi (hlm. 74).

### **2.3.1.3. Offer Informative Feedback**

Shneiderman dan Plaisant (2005), menyatakan bahwa setiap interaksi *user* dengan sistem harus memiliki umpan balik atau *feedback*. Sehingga *user* dapat mengetahui dengan jelas apa efek dari interaksi yang *user* lakukan. *Feedback* yang diberikan bisa saja kecil untuk interaksi yang minor, sebaliknya untuk interaksi yang major harus menggunakan *feedback* yang besar. *Feedback* dari sistem ini dapat berupa visual yang terlihat jelas (hlm. 74).

### **2.2.1.4. Design Dialogs to Yield Closure**

Urutan interaksi yang dapat digabungkan menjadi bagian awal, tengah, dan akhir. Dengan adanya pengelompokan interaksi tersebut dapat memberikan *feedback* yang informatif. *User* dapat merasakan rasa

keberhasilan serta lega setelah mendapatkan *feedback* yang informatif tersebut. Sebagai contoh adalah aplikasi *e-commerce* yang memberikan berbagai *feedback* informatif kepada *user* dalam proses pembelian produk. Dari memasukkan produk ke keranjang hingga notifikasi transaksi yang sukses (Shneiderman & Plaisant, 2005, hlm. 75).

#### **2.3.1.5. Prevent Errors**

Merancang desain yang dapat membimbing agar terhindar dari *error*. Misalnya seperti merancang sistem yang tidak memperbolehkan adanya alfabet saat diminta pengisian yang hanya berisi nomor. Serta desain yang tidak membingungkan *user* saat terjadi *error*. Pada saat terjadi *error*, *Interface* harus dapat mendeteksi *error* tersebut dan memberi *feedback* berupa instruksi yang simpel namun spesifik untuk *user*. Misalnya, pada saat menu *login*, *user* yang salah memasukkan *password* tidak harus mengetik ulang *email*. Sistem akan langsung memberitahu bahwa yang salah adalah *password*nya bukan *email*. Sehingga *user* tidak akan kebingungan (Shneiderman & Plaisant, 2005, hlm. 75).

#### **2.3.1.6. Permit Easy Reversal of Actions**

Menurut Shneiderman dan Plaisant (2005), interaksi dalam *Interface design* harus dapat diputar balik atau memiliki fitur kembali. Fitur ini membantu *user* untuk menjadi lebih santai dalam melakukan interaksi. *User* juga akan lebih bebas dalam memilih berbagai opsi yang tidak familiar (hlm. 75).

#### **2.3.1.7. Support Internal Locus of Control**

*User* yang telah memiliki pengalaman teknologi cenderung berkeinginan untuk menjadi inisiator di *Interface design*. Sehingga, desain yang menyulitkan *user* akan membangun ketidakpuasan dan kecemasan. Maka dari itu, desain harus dirancang untuk mendukung *user* menjadi

inisiator interaksi dibandingkan responden dari interaksi (Shneiderman & Plaisant, 2005, hlm. 75).

#### **2.3.1.8. Reduce Short-term Memory Load**

*Interface Design* perlu untuk dibuat simpel, dengan beberapa tampilan yang digabung, tampilan yang tidak terlalu banyak animasi, serta menyediakan waktu agar *user* mengingat kode dari *Interface*. Hal ini dikarenakan manusia memiliki batasan saat memproses ingatan yang jangka pendek (Shneiderman & Plaisant, 2005, hlm. 75).

#### **2.3.2. Icon Design**

Menurut Landa (2014) dalam merancang *icon* harus memikirkan beberapa hal penting. Hal tersebut berupa target audiens, ukuran *icon*, bagaimana dan apa konteks ketika *icon* ditampilkan di layar, tujuan komunikasi, hubungan *icon* dan sistem, serta gaya desain *icon* seperti apa yang cocok untuk proyek secara keseluruhan. Apalagi untuk *icon* aplikasi dan *website* yang mempunyai fungsionalitas dan desain solusi sendiri. Sehingga, *icon* tersebut harus memiliki korelasi yang besar dengan proyek. Hal ini dilakukan agar mencapai fungsionalitas yang baik, komunikasi yang tercapai, dan masih sambung dengan konteks proyek (hlm. 118-120).

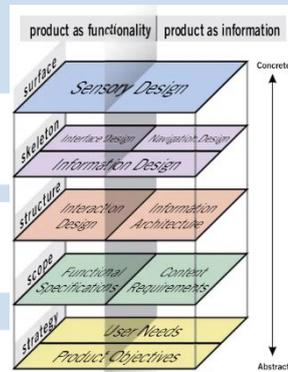
#### **2.4. User Experience**

Garrett (2011) menyatakan bahwa *user experience* adalah proses desain yang memastikan *user* mengalami pengalaman yang telah desainer ketahui dan perkiraan. Sehingga sebagai seorang desainer harus memikirkan segala langkah-langkah dan interaksi yang mungkin akan dilakukan *user* (hlm. 19).

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

## 2.4.1. Elemen *User Experience*

Menurut Garrett (2011), ada 5 macam elemen dalam merancang *user experience* yaitu (hlm. 20-30) :



Gambar 2.45. Elemen *User Experience*

Sumber: Garrett (2011)

### 2.4.1.1. *The Surface Plane*

Pada tampilan awal atau permukaan *website* dan aplikasi tersusun dari gambar-gambar ataupun teks yang interaktif dan dapat diklik. Sehingga, pada elemen ini berfokus pada pengalaman interaksi *user* terhadap dari produk final. Produk final pada permukaan awal cenderung berbentuk ilustrasi dan fotografi.

### 2.4.1.2. *The Skeleton Plane*

Elemen ini berfokus pada struktural dari tampilan site atau aplikasi. Sehingga *user* mendapatkan pengalaman yang konsisten dalam penempatan *button*, *control*, foto, maupun rangkaian teks. Dengan susunan yang terstruktur tentunya akan memberikan pengalaman yang baik dan efisien untuk *user*. Ada 3 komponen pada elemen *skeleton* yaitu desain informasi, desain *Interface*, dan desain navigasi, Ketiga komponen ini memastikan *user* mengalami kepastian, interaktif, dan merasa terarah.

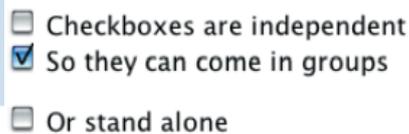
#### 1. Desain *Interface*

Dalam mendesain *Interface* ialah tentang memilih elemen *Interface* yang tepat untuk tugas yang tepat juga. Sehingga *user* dapat mengerti dan menggunakan elemen *Interface* dengan mudah.

*Interface* yang sukses adalah *Interface* yang dapat langsung memberitahu *user* tentang hal mana yang penting. Berikut beberapa contoh elemen *Interface* yang digunakan :

a. *Checkboxes*

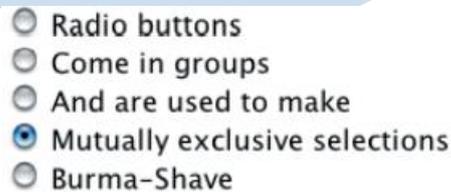
*Checkboxes* memberikan *user* pilihan untuk memilih 1 atau lebih pilihan.



Gambar 2.46. *Checkboxes*  
Sumber: Garrett (2011)

b. *Radio Buttons*

Elemen ini memberikan *user* memilih 1 pilihan di antara pilihan-pilihan yang eksklusif.



Gambar 2.47. *Radio Buttons*  
Sumber: Garrett (2011)

c. *Text Fields*

*Text fields* memperbolehkan *user* untuk mengetik di tempat yang telah diberikan.



Gambar 2.48. *Text Fields*  
Sumber: Garrett (2011)

#### d. *Dropdown Lists*

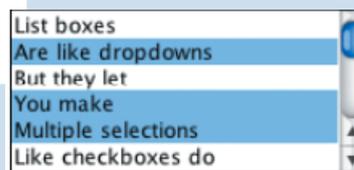
Elemen ini berfungsi seperti *radio button* tetapi memakan lebih sedikit tempat dibandingkan *radio button*. Dikarenakan hal itu, jawaban di *dropdown lists* dapat lebih beragam.



Gambar 2.49. *Dropdown Lists*  
Sumber: Garrett (2011)

#### e. *List Boxes*

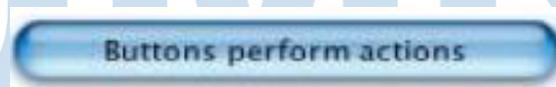
*List boxes* berfungsi seperti *checkboxes* tetapi menggunakan tempat yang lebih sedikit karena menggunakan fitur *dropdown*. Fitur tersebut membuat *List Boxes* dapat memuat lebih banyak opsi.



Gambar 2.50. *List Boxes*  
Sumber: Garrett (2011)

#### f. *Action Buttons*

Elemen ini memiliki banyak fungsi yang berbeda. Pada intinya, *button* ini berguna untuk memberitahu sistem tentang pilihan *user* dari elemen *Interface* lain. Elemen ini juga berguna untuk melakukan sesuatu, jadi bila di klik akan ada *action* yang muncul.



Gambar 2.51. *Action Buttons*  
Sumber: Garrett (2011)

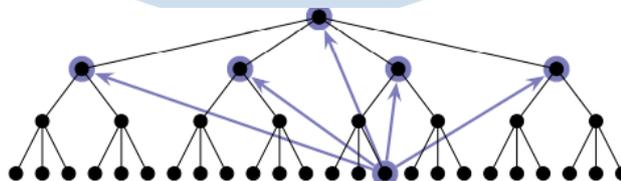
## 2. Desain Navigasi

Dalam merancang navigasi suatu aplikasi atau *site* harus memiliki 3 tujuan. Pertama, navigasi harus dapat menyediakan makna untuk *user* berpindah dari satu halaman ke halaman lain. Kedua, navigasi harus dapat memberikan informasi dan hubungan

dari setiap elemen di dalam navigasi tersebut. Sehingga *user* dapat mengetahui hubungan antar *link*/menu, menu yang lebih penting, dan perbedaan antar menu tersebut. Ketiga, desain navigasi harus dapat mengomunikasikan hubungan antara konten dan halaman yang sekarang sedang dilihat oleh *user*. Dengan memberitahu hal ini, *user* akan lebih terarah dalam mencapai tujuan yang mereka cari. Berikut beberapa jenis navigasi menurut Garret (2011):

a. *Global Navigation*

*Global Navigation* memberikan akses yang luas di setiap halaman aplikasi. Jadi, ada elemen yang selalu muncul di setiap halaman aplikasi. Sehingga menyatukan satu titik akses pengguna sehingga tidak perlu berpindah dari satu halaman ujung ke halaman ujung lainnya.

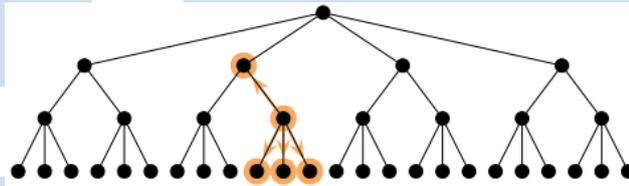


Gambar 2.52. *Global Navigation*  
Sumber: Garrett (2011)

UIN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

### b. *Local Navigation*

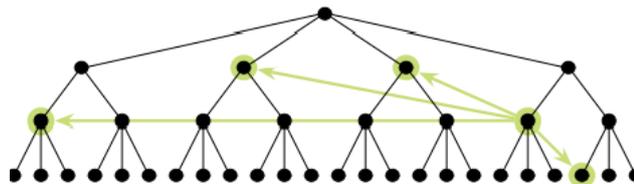
*Local navigation* memberikan akses kepada *user* mengenai hal-hal yang dekat / *nearby* di *Information Architecture/Wire flow*. Pada *Local Navigation* cenderung memberikan akses untuk cabang-cabang halaman.



Gambar 2.53. *Local Navigation*  
Sumber: Garrett (2011)

### c. *Supplementary Navigation*

Pada navigasi ini memberikan *shortcuts* ke konten yang tidak bisa diakses melalui navigasi global dan lokal. Hal ini memberikan kemungkinan untuk *user* mengalihkan fokus eksplorasinya tanpa mengulang dari halaman awal lagi. Navigasi ini akan tetap mempertahankan arsitektur hirarki utamanya.



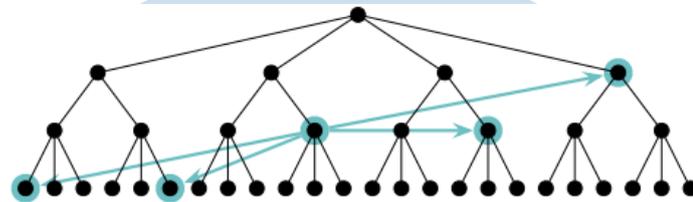
Gambar 2.54. *Supplementary Navigation*  
Sumber: Garrett (2011)

### d. *Contextual Navigation*

*Contextual navigation* adalah navigasi yang biasanya tertanam dalam konten halaman itu sendiri. Ketika *user* bingung akan suatu informasi dibanding memberikan *hyperlink* lebih baik

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

memberikan informasi tambahan di dalam halaman itu juga. Oleh karena itu dibutuhkan *contextual navigation*.

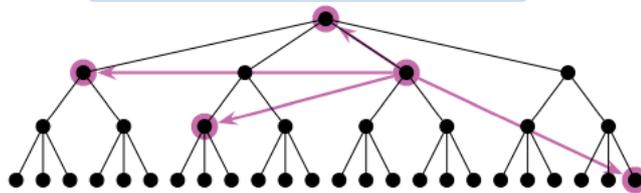


Gambar 2.55. *Contextual Navigation*

Sumber: Garrett (2011)

#### e. *Courtesy Navigation*

Pada navigasi ini memberikan akses ke berbagai informasi yang mungkin tidak diperlukan setiap hari tetapi tetap disediakan untuk kenyamanan *user*. Elemen-elemen yang biasa ada pada navigasi ini adalah forum *feedback*, nomor kontak, dan *policy statements*.

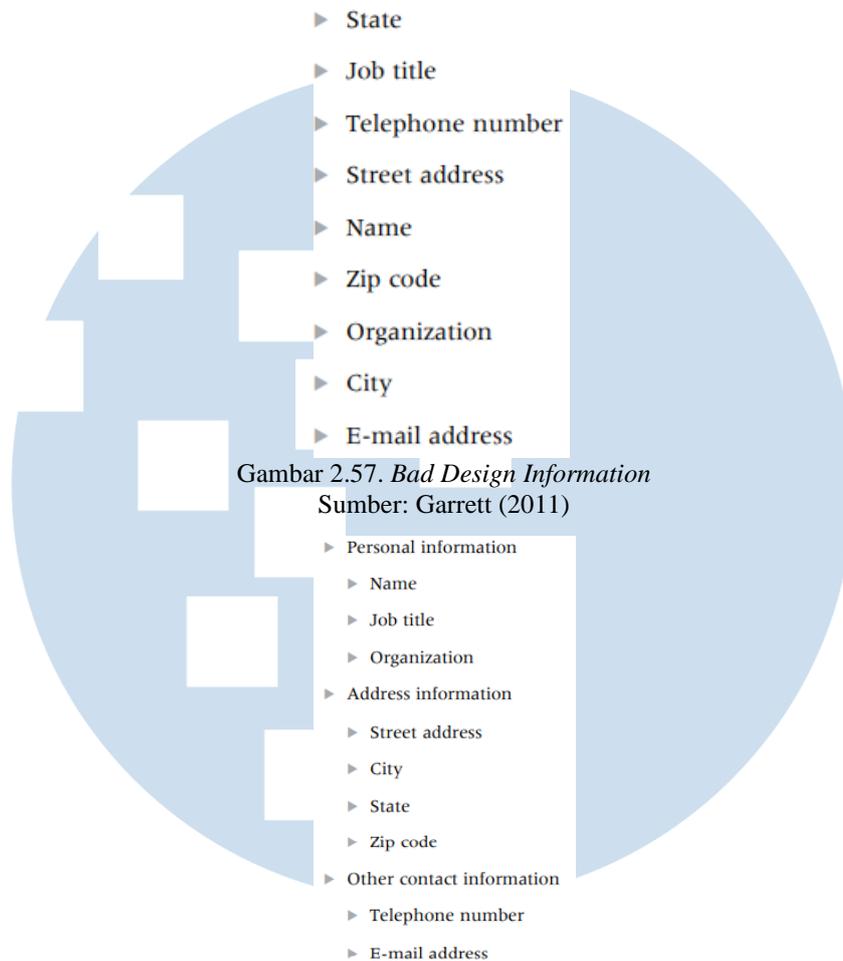


Gambar 2.56. *Courtesy Navigation*

Sumber: Garrett (2011)

### 3. Desain Informasi

Perancangan desain informasi sering dijadikan sebagai hal penting untuk menggabungkan dan mempertahankan setiap komponen-komponen desain. Dalam merancang desain informasi berhubungan dengan keputusan desainer dalam menampilkan informasi agar *user* dapat mengerti dan menggunakan informasi dengan mudah. Misalnya dalam memberikan informasi harus terstruktur dan kalau perlu dibagi sesuai kelompok dan kriteria.



Gambar 2.57. *Bad Design Information*  
 Sumber: Garrett (2011)

Gambar 2.58. *Good Design Information*  
 Sumber: Garrett (2011)

### 2.4.1.3. *The Structure Plane*

Berbeda dengan *skeleton* yang masih abstrak, elemen *structure* memberikan kejelasan proses kepada *user*. Misalnya *skeleton* memberikan penempatan *Interface* yang jelas pada saat *checkout e-commerce*, sedangkan *structure* memberikan gambaran tentang bagaimana *user* sampai ke tahap *asdstructure* memberikan informasi arsitektur dari *site* atau aplikasi tersebut. Sehingga informasi arsitektur memberikan fasilitas kepada *user* untuk mencapai pemahaman.

#### **2.4.1.4. The Scope Plane**

Rangkaian fitur dan *function* pada aplikasi. Misalnya seperti *e-commerce* yang memberikan fitur berupa penyimpanan alamat yang pernah dikirim. Namun, ada juga *e-commerce* yang tidak menyediakan fitur penyimpanan ini. Ada atau tidaknya fitur penyimpanan ini dalam suatu aplikasi adalah elemen *scope*.

#### **2.4.1.5. The Strategy Plane**

Fitur dan *function* dari elemen *scope* sangat bergantung dari fundamental yang disusun pada elemen *strategy* aplikasi. *Strategy* tidak hanya memikirkan apa yang ingin didapatkan desainer dari aplikasi yang dirancang, tetapi juga apa yang dibutuhkan dan diinginkan *user* dari aplikasi tersebut. Contohnya penjual pasti ingin menjual produk, sedangkan pembeli ingin membeli produk.

Oleh karena itu, kebutuhan *user* adalah fokus utama desainer dalam membuat aplikasi. Desainer harus mengerti apa yang *user* inginkan dan bagaimana menyesuaikan dengan keinginan *user*. Namun, dari tim desainer juga harus mempunyai *goal* atau *objective* dari aplikasi. Misalnya peningkatan penjualan di aplikasi.

### **2.4.2. User-centered design**

*User-centered design* memberikan pengalaman yang efisien kepada *user*. Dalam menciptakan pengalaman yang baik dan efisien sebenarnya gampang dipahami namun akan kompleks saat dilakukan. Dengan memikirkan setiap langkah yang mungkin akan dilakukan oleh *user* setiap desainer mengembangkan aplikasi atau produk. Segala pengalaman yang dirasakan *user* harus sudah menjadi kesadaran dari desainer. Memang tidak semua pengalaman yang buruk dapat tertutupi apabila *budget* tidak memenuhi. Namun, sebagai desainer harus menyadari segala keputusan *user* agar memastikan tidak adanya kecelakaan atau pengalaman yang sangat buruk (Garrett, 2011, hlm. 17).

Desainer harus memikirkan matang-matang *user experience* karena berhubungan langsung dengan *user*. Apabila desainer tidak berhasil memberikan layanan yang positif, maka *user* tidak akan kembali. Bila *user* tidak kembali berarti aplikasi yang dibuat hanya menjadi pajangan yang menunggu *user* untuk berkunjung kembali. Oleh karena itu, berikanlah *user* pengalaman terbaik saat berada di aplikasi dengan layanan yang nyaman, intuitif, dan memuaskan. Desainer juga harus memastikan segala hal dalam aplikasi berfungsi layak seharusnya (Garrett, 2017, Hlm. 17).

#### **2.4.3. Workflow**

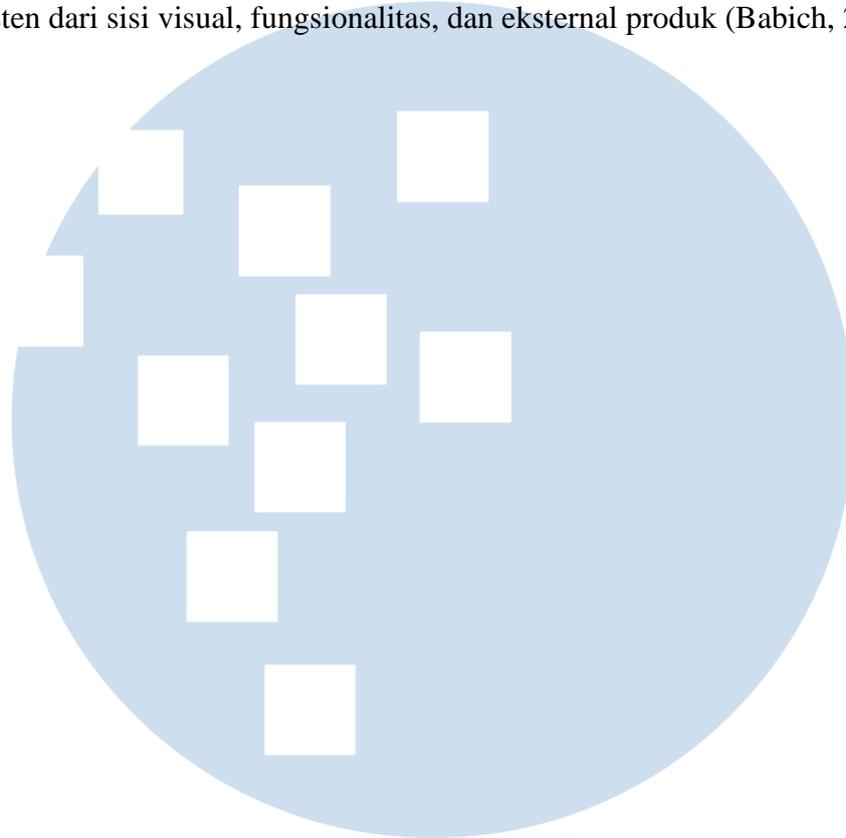
Cooper, dkk. (2007) menyatakan bahwa *workflow* berguna untuk mendapatkan alur informasi dan dalam membuat proses keputusan organisasi. *Workflow* biasa menggunakan *flowchart* atau diagram-diagram lain yang dapat menangkap berbagai fenomena. Fenomena tersebut berupa tujuan atau hasil proses, tingkat kepentingan suatu aksi, apa yang menginisiasi proses, tokoh-tokoh yang terlibat dan peran, aksi spesifik yang dilakukan, keputusan yang dibuat, informasi yang mendukung keputusan, kasus-kasus eror, serta bagaimana kasus eror tersebut diperbaiki (hlm. 106-107).

#### **2.5. Aplikasi**

Menurut Babich (2018), perbedaan dari aplikasi yang baik dan buruk terletak pada UX aplikasi tersebut. Alasan dibalik ketidaksuksesan aplikasi juga terletak pada UX masing-masing aplikasi. *User* mengharapkan tiga hal penting dari aplikasi yaitu gampang digunakan, *loading* aplikasi yang cepat, dan interaksi aplikasi yang memuaskan.

Dalam mendesain aplikasi harus dapat meringankan beban *user* dengan menerapkan tampilan yang minimalis dan dibutuhkan oleh *user* saja. Sistem aplikasi juga dapat didesain untuk mengingat data *user*. Pembagian proses menjadi tahap-tahap juga dapat meringankan beban *user* dan membuat tahap terkesan simpel. Menggunakan *Interface* yang familiar serta mengurangi berbagai *input*

untuk meringankan bebas *user*. Aplikasi juga harus didesain dengan konsisten. Konsisten dari sisi visual, fungsionalitas, dan eksternal produk (Babich, 2018).



# UMMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA