BAB II

TINJAUAN PUSTAKA

2.1 Desain

Menurut Richard Grefe dalam Landa (2014), desain merupakan pengantara dari informasi dan pemahaman yang didapatkan oleh seseorang. Menurut Landa (2014), Desain grafis sendiri adalah suatu bentuk komunikasi dengan elemen visual yang digunakan untuk menyampaikan pesan kepada penerima yang merepresentasikan ide berdasarkan kreasi, seleksi, dan juga gabungan dari sekelompok elemen visual. Desain komunikasi penting untuk digunakan karena desain komunikasi dapat membantu mendorong perekonomian, memberikan informasi-informasi kepada publik, dan mempromosikan kompetisi yang dapat berujung kepada riset perkembangan barang dan jasa.

2.1.1 Prinsip Desain

Prinsip desain digunakan untuk menyusun desain untuk menciptakan keseimbangan yang stabil dalam komposisi (Landa, 2014). Prinsip desain sendiri dibagi menjadi enam yaitu format, keseimbangan, hierarki visual, ritme, kesatuan, dan gestalt. Keenam prinsip itu saling berhubungan untuk membentuk suatu desain dengan komposisi yang baik.

2.1.1.1 Format

Format merupakan batasan luar dari sebuah desain. Format juga dapat mengacu pada bidang yang digunakan pada sebuah proyek desain grafis. Desainer juga menggunakan kata format sebagai jenis proyek yang dikerjakan seperti poster, buku, *website*, dan lain sebagainya. Setiap jenis format yang ada memiliki ukuran standar masing-masing yang harus diikuti. Tetapi terdapat juga format yang tidak memiliki batasan ukuran sehingga desainer dapat mengatur ukuran format dengan leluasa (Landa, 2014).

2.1.1.2 Keseimbangan

Keseimbangan merupakan prinsip yang mengatur stabilitas dan pendistribusian visual dan elemen-elemen yang ada pada komposisi dari satu sisi dan sisi lainnya. Keseimbangan pada desain akan menciptakan suatu harmoni yang dapat memberikan dampak positif bagi penonton. Prinsip keseimbangan ini tidak dapat berdiri sendiri karena prinsip ini berhubungan dengan prinsip-prinsip lainnya (Landa, 2014).

2.1.1.3 Hierarki visual

Visual hierarki merupakan prinsip yang digunakan untuk membimbing penonton kepada penyusunan elemen-elemen grafis berdasarkan empasis yang ada pada suatu desain. Empasis merupakan penyusunan elemen grafis menurut urutan kepentingan dengan menonjolkan elemen tertentu daripada elemen lainnya. Desainer akan menentukan setiap urutan elemen yang akan dilihat oleh penonton (Landa, 2014).

2.1.1.4 Ritme

Ritme pada desain grafis merujuk pada repetisi yang konsisten dari sebuat elemen yang dapat membuat mata penonton untuk melihat karya desain secara keseluruhan. Tempo ritme dapat diatur melalui jarak maupun posisi antar elemen pada sebuah karya. Ritme merupakan prinsip penting dalam pembuatan visual yang saling bersambungan dan berkaitan seperti dalam pembuatan halaman-halaman pada sebuah buku. Ritme dapat dipengaruhi oleh beberapa faktor seperti warna, tekstur, bentuk, empasis, dan keseimbangan (Landa, 2014).

2.1.1.5 Kesatuan

Kesatuan merupakan suatu prinsip desain yang dapat membuat setiap elemen di dalam desain menjadi satu kesatuan dan terlihat tidak dapat terpisah. Komposisi dengan kesatuan yang baik

juga dapat membuat penonton lebih mudah memahami dan mengingat komposisi. Kesatuan dapat dicapai pada desain dengan mengkaitkan setiap elemen yang ada pada desain dengan satu sama lainnya (Landa, 2014).

2.1.1.6 Hukum gestalt

Menurut Landa (2014), Terdapat enam hukum yang ada di hukum gestalt yaitu:

1) *Similarity* yang menandakan elemen yang memiliki kesamaan karakteristik dan dapat dipersepsikan sebagai satu bagian. Kesamaan elemen dapat dilihat dari bentuk, tekstur, warna, maupun arah.

2) *Proximity* menandakan jarak antar elemen yang saling berdekatan sehingga dapat dianggap sebagai suatu kelompok elemen.

3) *Continuity* merupakan koneksi atau laju sebuah visual yang terlihat bersambungan dan berkaitan sehingga dapat memberikan kesan pergerakkan.

4) *Closure* merupakan kecenderungan pemikiran untuk menyambungkan elemen untuk dapat menghasilkan bentuk yang sempurna atau komplit.

5) *Common fate* merupakan gabungan dari elemen yang dianggap sebagai bagian dari sebuah unit karena mempunyai pergerakan dengan arah yang sama.

6) *Continuing line* merupakan persepsi secara garis besar dimana kumpulan elemen akan dilihat secara keseluruhan daripada secara individu.

2.1.2 Visual dalam Aplikasi

Tampilan dari sebuah aplikasi terdiri dari berbagai elemen seperti tombol, grafik, *icon,* dan *background*. Setiap elemen pada aplikasi memiliki karakteristik visualnya mulai dari warna, ukuran, dan *font* yang disusun untuk membuat estetika aplikasi (Cuello dan Vittone, 2013).

2.1.2.1 Identitas Visual

Aplikasi merupakan alat komunikasi yang dapat membantu menyebarluaskan identitas sebuah perusahaan. Identitas perusahaan dapat ditunjukkan dengan warna, *font*, dan *background* dari sebuah aplikasi. Ada baiknya pengenalan identitas sebuah *brand* tidak terlalu mempengaruhi navigasi dan *user experience* dari aplikasi. Hal tersebut dapat dihindari dengan beberapa cara seperti meletakkan *brand* pada halaman pengenalan atau halaman awal maupun halaman khusus mengenai *brand* tersebut (Cuello dan Vittone, 2013).

2.1.2.2 *Icon*

Menurut Cuello dan Vittone (2013), sebuah aplikasi harus dapat memberikan impresi yang baik dari pertama kali seorang pengguna melihat visual aplikasi tersebut. Impresi pertama yang dilihat oleh pengguna berasal dari dua komponen yaitu *launch icon* dan *launch image* yang muncul ketika aplikasi dibuka.

1) ***Launch icon***

   Icon ini merupakan visual yang merepresentasikan aplikasi pada perangkat pengguna. Oleh karena itu *launch icon* harus dapat menonjol dari aplikasi lain dan merepresentasikan aplikasi dengan baik. *Icon* yang efektif dapat dibuat dengan menggunakan bentuk sederhana dan rapih.

2) ***Interior icon***

   Interior icon merupakan *icon* yang terletak di dalam aplikasi. *Icon* ini digunakan sebagai alat bantu visual dalam penyampaian

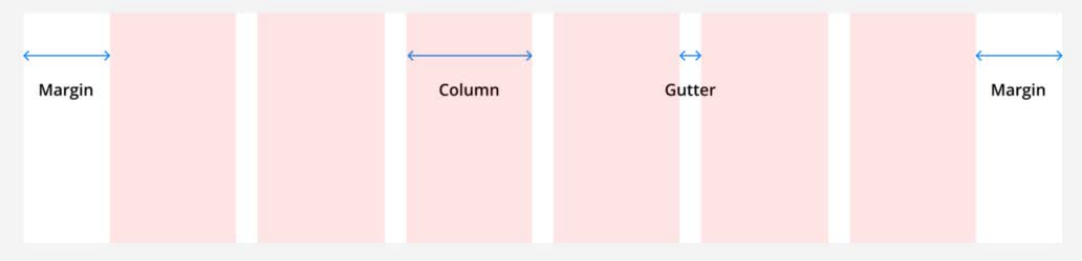
informasi. *Icon* harus dapat menyampaikan aksi yang digambarkan sesuai dengan konteksnya. *Icon* harus jelas dan representatif khususnya untuk *icon* yang berdiri sendiri tanpa teks informasi.

3) ***Launch image***

Launch image atau yang bisa disebut juga dengan *splash*, merupakan layar pertama yang akan dilihat oleh pengguna ketika membuka aplikasi. Biasanya layar ini ditunjukkan secara singkat ketika aplikasi pertama kali dibuka. Karena hanya ditunjukkan untuk waktu yang singkat, informasi yang ditunjukkan dibatasi dengan nama dan versi aplikasi.

2.1.2.3 Grid

Grid merupakan sebuah pedoman dalam penyusunan elemen visual yang terdiri dari struktur vertikal dan horizontal yang membentuk kolum dan margin. Grid dapat menjadi kerangka yang membantu penyusunan komposisi untuk setiap halaman desain yang dibuat sehingga desain yang dibuat memberikan kesan keterlanjutan, kesatuan, dan alur visual yang sama dalam berbagai media (Landa, 2014).



Gambar 2.1 Anatomi Grid
Sumber: https://medium.com/design-with-figma/everything-you-need-to-know-as-a-ui-designer-about-spacing-layout-grids-2bc269e12321, 2019

Grid dapat membantu desainer dalam membuat halaman, baik secara digital maupun cetak. Dalam pembuatan desain konten yang tediri dari banyak halaman, maka setiap struktur yang ada didalamnya harus dapat mudah diakses dan dibaca. Maka dari itu,

grid dapat membantu untuk memberikan kesatuan dan keseimbangan dalam setiap halaman.

1) Margin

Margin merupakan ruang kosong yang ada pada sisi kanan, kiri, atas, maupun bawah sebuah desain. Margin memiliki fungsi sebagai struktur proporsional disekitar konten visual sebuah desain. *Margin* yang ada pada sebuah desain dapat membantu desainer untuk menentukan jarak antara gambar maupun teks sehingga setiap elemen tetap berada di dalam format yang ditentukan (Landa, 2014).

Menurut Poulin (2018), proporsi dari sebuah *margin* sangat penting untuk dipertimbangkan karena *margin* dapat membantu untuk menentukan keseimbangan sebuah desain secara keseluruhan dan juga secara komposisi *layout*. *Margin* juga dapat digunakan desainer untuk memberikan fokus dan atensi pengguna.

2) Kolom

Kolom adalah susunan vertical yang dibuat untuk mengakomodasi teks maupun gambar. Dalam setiap grid, jumlah kolom yang digunakan bervariasi bergantung dengan konsep, tujuan, maupun cara pemaparan konten yang ingin dibuat oleh desainer. Ketika desainer menggunakan kolom lebih dari satu, maka setiap kolom dapat memiliki ukuran yang sama maupun beragam. Setiap kolom dapat digunakan menjadi tempat teks atau gambar maupun gabungan dari keduanya. Jarak yang terbuat diantara kolom disebut sebagai interval kolom (Landa, 2014).

Penentuan panjang dan lebar kolom penting dalam penyesuaian keterbacaan yang optimal. Ukuran tulisan, panjang garis dan

leading juga menjadi karakteristik penting dalam penentuan ukuran sebuah kolom. Dalam penentuan ukuran teks, pertimbangan jumlah kata yang dapat diakomodasi oleh kolom juga penting dalam memastikan pengguna dapat membaca dengan mudah (Poulin, 2018).

3) Gutter

Gutter adalah ruang kosong yang digunakan untuk memisahkah jarak antara dua kolom. Selain memberikan pemisahan kolom secara visual, lebar dan proporsi dari *gutter* dapat memberikan dampak kepada tampilan, komposisi visual, dan keterbacaan secara keseluruhan sebuah desain. Selain menjadi pemisah antara elemen halaman, *gutter* juga dapat menjadi kunci elemen visual dalam sebuah komposisi. Dengan meminimalisasi atau memaksimalkan lebar *gutter* sesuai dengan elemen visual yang ada.

Penggunaan variasi *gutter* juga dapat memberikan keseimbangan atau memberikan penekanan terhadap setiap elemen yang ada pada desain. Fokus terhadap suatu elemen, pemisahan area untuk informasi sekunder, serta penyesuaian komposisi juga dapat dilakukan dengan penggunaan *gutter* yang tepat.

4) Jenis *Grid*

Grid memiliki beberapa macam jenis berdasarkan penggunaan kolom. Setiap jenis kolom juga memiliki kelebihan masing-masing dan penggunaannya dapat beragam. Dalam penggunaan jenis *grid* dalam sebuah desain dapat ditentukan melalui tujuan, ukuran konten tulisan dan gambar, dan target sasaran (Landa, 2014).

a) *Grid* kolom tunggal

Grid ini merupakan jenis *grid* yang paling mendasar dan sering juga disebut sebagai *manuscript grid*. *Grid* ini memiliki satu kolom teks yang dikelilingi oleh margin. Biasanya grid dengan jenis ini cocok digunakan untuk layar berukuran kecil yang ada pada layar *smartphone*.

b) Grid multi kolom

Kolom yang ada pada *grid* kolom tunggal dapat dipecah menjadi beberapa kolom secara simetris maupun asimetris. Pemecahan kolom ini kemudian disebut sebagai *grid* multi kolom. *Grid* multi kolom yang biasa digunakan untuk *desktop, tablet,* dan layar *mobile* didesain dan diukur dengan menggunakan pixel.

c) *Modular Grid*

Module adalah unit yang terbuat dari pertemuan antara kolom vertikal dan garis horizontal. Sebuah *module* dapat diisi dengan teks atau gambar. Setiap teks maupun gambar juga dapat diletakkan dalam beberapa modul. *Modular grid* adalah grid yang tersusun dari beberapa modul. *Modular grid* memiliki keuntungan dalam mengelompokkan informasi ke dalam masing-masing modul. *Modular grid* juga merupakan *grid* yang paling fleksibel sehingga dapat dimanfaatkan untuk membuat variasi yang beragam dalam sebuah desain.

2.1.2.4 Tipografi

Tujuan dari tipografi adalah membuat teks mudah dibaca oleh pengguna. Tujuan ini dapat dicapai dengan penggunaan jenis *font*, penyesuaian ukuran, *spacing,* dan kontras visual *background.* Kontras merupakan hal yang penting diperhatikan dalam aplikasi karena keterbacaan *smartphone* saat berada di luar ruangan bergantung pada kontras antara *typeface* dan *background*. Tipografi

merupakan sebuah komponen yang terletak pada *grid* (Cuello dan Vittone, 2013).

1) ***Serif* dan *sans-serif***

Pada layar yang memiliki ukuran dan resolusi kecil, tipografi yang disarankan untuk digunakan adalah tipografi *sans-serif*. Tipografi serif dapat digunakan untuk bagian judul.

2) **Ukuran minimal**

Ketika menggunakan tipografi dengan ukuran kecil, desainer harus dapat memilih *font* sederhana yang memiliki jarak *line* dan *spacing* untuk memberikan perpisahan secara visual sehingga dapat mudah dibaca. Ukuran *font* yang biasanya digunakan pada *smartphone* Android berukuran 12sp hingga 22 sp.

3) **Hierarki**

Hierarki dalam tipografi penting digunakan untuk menyusun informasi dan posisi setiap konten pada layar. Untuk memisahkan tingkat hierarki, desainer dapat menggunakan perbedaan ukuran, variabel *font,* dan warna.

2.1.2.5 Warna

Warna merupakan sebuah aset penting dalam aplikasi. Menurut Cuello dan Vittone (2013), Warna digunakan pada *heading,* teks, tombol, *background,* dan elemen lainnya yang ada pada *interface* aplikasi. Warna pada aplikasi dapat diasosiasikan dengan identitas *brand* atau dengan pemilihan estetika desain.

1) ***Reserved colors***

Beberapa warna harus digunakan secara hati-hati karena konotasi yang ada di balik warna tersebut. Beberapa contoh warna yang harus diperhatikan adalah:

a) Merah: digunakan untuk *error* dan peringatan.
b) Kuning: digunakan untuk peringatan.
c) Hijau: digunakan untuk memberikan pesan sukses atau berhasil.

2) Teks

Warna pada teks dapat membantu dalam penyusunan hierarki teks. Teks dapat diberikan fokus maupun dikurangi fokusnya bergantung dengan warna yang diberikan.

3) Latar belakang

Warna latar belakang harus sejalan dengan warna teks untuk dapat memberikan kontras untuk memberikan tingkat keterbacaan yang baik.

4) Elemen interaktif

Elemen interaktif seperti tombol maupun baris dapat disorot dengan menggunakan warna indikasi aksi pengguna. Pada elemen yang tidak dapat diinteraksikan, biasanya warna yang diberikan akan lebih pudar dari warna aslinya.

2.1.2.6 Garis

Titik merupakan satuan terkecil dari elemen garis. Titik yang dipanjangkan akan menjadi sebuah garis. Garis sendiri merupakan sebuah tanda yang dibuat dari alat visualisasi yang digunakan pada sebuah permukaan. Garis dapat dibuat dengan berbagai macam alat seperti pensil, kuas, *stylus,* dan berbagai macam alat lain yang dapat memberikan tanda pada suatu permukaan. Garis pada umumnya diidentifikasi melalui panjang garis tersebut dibandingkan lebarnya (Landa, 2014).

2.1.2.7 Bentuk

Bentuk merupakan gambaran garis besar dari sesuatu. Bentuk biasanya dibentuk dari permukaan dua dimensi yang dibuat dari sebagian atau seluruhnya dari garis atau warna atau tekstur (Landa, 2014). Bentuk bersifat dua dimensional yang dapat diukur dari panjang dan lebar. Semua bentuk yang ada berasal dari tiga penggambaran dasar yaitu kotak, segitiga, dan lingkaran yang juga memiliki wujud volumetric yaitu kubus, pyramid, dan bola.

2.1.2.8 Tekstur

Tekstur adalah sensasi yang dirasakan dengan rabaan atau sentuhan yang merepresentasikan suatu permukaan. Terdapat dua kategori yang ada pada seni visual yaitu rabaan dan visual. Tekstur rabaan merupakan tekstur yang dapat dirasakan dan diraba secara langsung. Sedangkan tekstur visual merupakan tekstur yang dibuat dari teknik percetakan yang menghasilkan tekstur pada desain cetak (Landa, 2014).

2.2 Aplikasi

Menurut Cuello dan Vittone (2013), aplikasi merupakan sebuah perangkat lunak yang dapat digambarkan sebagai program yang ada pada *smartphone*. Pada awalnya aplikasi difokuskan untuk meningkatkan produktivitas individu seperti aplikasi *alarm*, kalender, kalkulator, maupun *email*. Sekarang aplikasi berkembang menjadi sebuah peluang bagi pengembang aplikasi dalam mencari keuntungan dalam *marketplace* seperti App Store maupun Play Store.

2.2.1 Jenis Aplikasi

Menurut Cuello dan Vittone (2013), Aplikasi dapat dikembangkan dengan banyak cara. Perbedaan cara pengembangan inilah kemudian dapat memunculkan beragam karakteristik dan batasan pada sebuah aplikasi. Jenis aplikasi yang dibuat dapat bergantung dengan jenis interaksi dan desain yang dibuat.

2.2.1.1 *Native Applications*

Aplikasi ini merupakan aplikasi yang dirancang bersamaan dengan *operating system* yang biasa disebut sebagai *Software Development Kits* (SDKs). *Smartphone* Android, iOS, dan Windows masing-masing memiliki SDKnya tersendiri dan *native application* yang tersedia berbeda dan didesain khusus untuk setiap *platform* (Cuello dan Vittone, 2013).

Aplikasi dengan jenis ini merupakan aplikasi yang harus diunduh terlebih dahulu dan di *install* ke *smartphone.* Aplikasi dengan jenis ini juga memiliki pembaruan secara berkala sehingga pengguna harus mengunduh aplikasi ini secara terus-menerus untuk mendapatkan versi aplikasi yang paling baru. Karakteristik lainnya dari aplikasi jenis ini adalah aplikasi ini dapat menggunakan notifikasi peringatan bawaan dari *operating system* yang dimiliki oleh *smartphone* walaupun aplikasi sedang tidak digunakan. Selain itu, aplikasi ini tidak membutuhkan koneksi internet sehingga aplikasi ini dapat memberikan pengalaman yang lebih lancar dan terintegrasi dengan *smartphone.* Biasanya aplikasi ini menggunakan *hardware* yang ada pada *smartphone* seperti kamera dan sensor.

Gambar 2.2 Notifikasi Peringatan
Sumber: Cuello dan Vittone (2013)

Secara desain, *native application* memiliki tampilan yang mengikuti *guideline* dari masing-masing *operating system* sehingga setiap aplikasi menjadi konsisten. Tampilan ini juga dapat bermanfaat untuk kemudahan dalam penggunaan aplikasi bagi pengguna.

2.2.1.2 *Web Applications*

Web application merupakan aplikasi yang diprogram dengan menggunakan HTML, JavaScript, dan CSS. Pada aplikasi ini, SDKs tidak digunakan sehingga aplikasi dapat berjalan secara independen dari *operating system smartphone*. Hal ini juga merupakan alasan kenapa aplikasi ini dapat digunakan di berbagai *platform* tanpa harus dirancang secara terpisah (Cuello dan Vittone, 2013).

Aplikasi ini juga tidak perlu diunduh oleh pengguna karena aplikasi ini dapat diakses melalui *browser* seperti *website* pada umumnya. Karena aplikasi ini dapat diakses melalui *web*, maka setiap pengguna tidak perlu memperbarui aplikasi ini. Akan tetapi, pengguna harus menggunakan koneksi internet untuk dapat mengakses aplikasi ini.

Gambar 2.3 *Web Application*
Sumber: Cuello dan Vittone (2013)

Tampilan yang dimiliki aplikasi ini cenderung generik dan terpisah dari tampilan *operating system* tiap *smartphone.* Identifikasi pengguna dan elemen navigasi maupun interaksi biasanya tidak terlalu signifikan pada aplikasi jenis ini.

2.2.1.3 *Hybrid Applications*

Aplikasi ini merupakan gabungan dari kedua jenis aplikasi sebelumnya. Secara pengembangan, aplikasi ini menyerupai pengembangan aplikasi web yang menggunakan HTML, JavaScript, dan CSS. Setelah aplikasi selesai dikembangkan, maka aplikasi akan di susun sehingga hasil akhirnya serupa dengan *native application*. Keuntungan dari aplikasi jenis ini adalah aplikasi akan dapat menggunakan kapasitas penyimpanan *smartphone* seperti *native application*. Tampilan aplikasi ini tidak terlalu bergantung dengan *operating system* dari *smartphone*, tetapi desainer dapat menggunakan desain tombol dan kontrol *native* untuk memberikan kesan estetika tertentu (Cuello dan Vittone, 2013).

2.2.2 Kategori Aplikasi

Berdasarkan konten yang ditampilkan kepada pengguna, aplikasi dibagi menjadi beberapa kategori. Setiap kategori aplikasi dapat menentukan seberapa kompleks tampilan aplikasi. Beberapa aplikasi juga dapat termasuk dalam lebih dari satu kategori (Cuello dan Vittone, 2013). Berikut adalah kategori aplikasi:

1) ***Entertainment***

   Kategori ini biasanya digunakan untuk aplikasi *game* dan aplikasi untuk bersenang-senang. Tampilan aplikasi kategori ini biasanya menggunakan efek grafik, animasi, dan suara untuk mengikat perhatian pengguna.

2) ***Social***

   Aplikasi yang ada pada kategori ini lebih terfokuskan pada komunikasi, kontak, dan interaksi antara pengguna. Contoh aplikasi dalam kategori ini adalah Facebook, Instagram, dan lainnya.

3) ***Utilities and productivity***

Kategori ini sering disambungkan dengan sektor bisnis dimana aplikasi ini menyediakan alat untuk menyelesaikan masalah spesifik berdasarkan eksekusi tugas yang memprioritaskan efisiensi. Aplikasi dengan kategori ini mencangkup aplikasi *to-do list*, aplikasi kerja tim, dan lainnya.

4) ***Educational and informative***

Aplikasi kategori ini digunakan untuk menyampaikan informasi, pengetahuan, dan berita. Hal yang paling utama dalam kategori ini adalah akses ke konten sehingga keterbacaan dan navigasi informasi yang ada di aplikasi harus di perhatikan dengan baik.

5) ***Creation***

Aplikasi ini merupakan aplikasi yang berpusat pada kreativitas pengguna. Contoh aplikasi dengan kategori ini adalah aplikasi untuk mengedit video, foto, maupun suara.

2.2.3 Perancangan Aplikasi

Perancangan sebuah aplikasi dimulai dari sebuah ide yang kemudian di analisa hingga terbentuk sebuah aplikasi yang dapat dipasarkan. Dalam berbagai tahapan, desainer bekerja sama dengan *developer* untuk saling berkonsultasi terkait proses pembuatan aplikasi. Dalam perancangan aplikasi, desainer perlu memperhatian koordinasi setiap pihak yang terlibat dan menggunakan setiap metode yang ada untuk menghasilkan aplikasi yang dapat digunakan (Cuello dan Vittone, 2013).

2.2.3.1 Persona

Persona merupakan sebuah alat yang digunakan dalam pembuatan desain interaksi. Fungsi dari *persona* adalah untuk menentukan model pengguna dan struktur dasar desain. Untuk dapat membuat *persona,* dibutuhkan riset untuk menganalisa peilaku dan pola pikiran yang dimiliki oleh beragam pengguna sehingga desainer

dapat menemukan kesamaan diantara setiap user (Cuello dan Vittone, 2013).

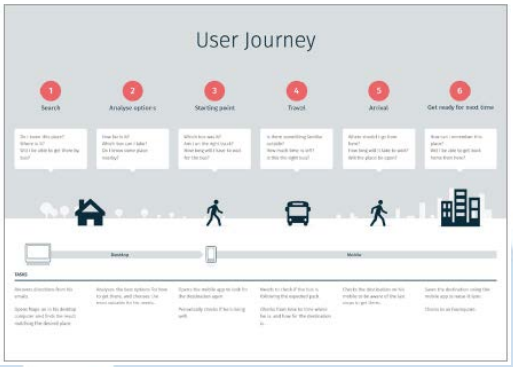
Gambar 2.4 Contoh *Persona*
Sumber: Cuello dan Vittone (2013)

Setelah dilakukannya riset, maka desainer akan mendapatkan representasi visual untuk membentuk sebuah *persona.* Untuk membuat aplikasi, desainer dapat membuat bermacam-macam *persona*, tetapi sebaiknya persona dibuat tidak lebih dari tiga. Secara ideal persona yang harus difokuskan hanya satu saja.

2.2.3.2 User Journey

Setelah pembuatan *persona*, desainer dapat memahami perilaku pengguna dalam mencapai tujuan dalam konteks yang diberikan (Cuello dan Vittone, 2013). Untuk dapat memahami perilaku tersebut, desainer dapat menggunakan *user journey* sehingga setiap proses yang dilakukan oleh *persona* dapat tervisualisasikan.

UNIVERSITAS MULTIMEDIA NUSANTARA

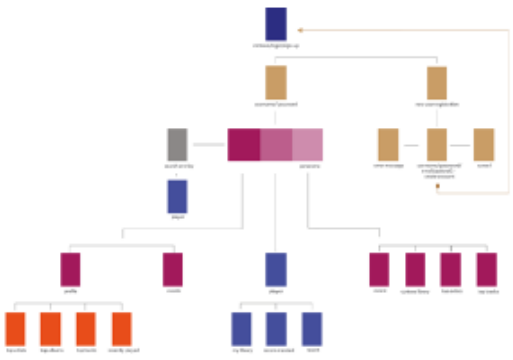


Gambar 2.5 Contoh *User Journey*
Sumber: Cuello dan Vittone (2013)

Pembuatan *user journey* ini dapat divisualisasikan dengan grafik secara linear. Emosi dan kesulitan yang dihadapi oleh pengguna dapat digambarkan pada setiap bagian tahapan yang dilalui. Dengan adanya pembagian tahapan ini, maka desainer dapat melihat bagian desain yang perlu difokuskan dan memperbaiki mas

2.2.3.3 *Information Architecture*

Information architecture merupakan sebuah diagram yang digunakan untuk menyusun setiap konten dan fitur yang ada pada aplikasi sehingga dapat ditemukan dengan mudah dan cepat. *Information architecture* ini juga dapat menjadi penentu layar dan fitur yang akan dibutuhkan pada setiap tahapan. Dalam pembuatannya, biasanya digambarkan dengan kotak sebagai layar dan garis panah menjadi penunjuk arah navigasi antar layar (Cuello dan Vittone, 2013).



Gambar 2.6 Contoh *Information Architecture*
Sumber: Cuello dan Vittone (2013)

Diagram ini dapat membantu untuk melihat seberapa kompleks aplikasi yang dibuat dan membantu untuk menganalisa kedalaman dan relasi setiap konten dengan rapih. Hasil dari diagram ini akan membantu dalam pemilihan navigasi aplikasi. *Information architecture* dibagi menjadi tiga komponen yaitu *ontology, taxonomy,* dan *choreography* ("Understanding Information Architecture", n.d.). komponen tersebut berperan untuk memastikan *information architecture* yang dibuat berguna, relevan, dan otentik.

1) Ontology

Ontology merupakan kumpulan makna yang dapat dimengerti oleh banyak orang. Menurut Simon (2015), pemilihan *ontology* tidak hanya didasari oleh pemilihan bahasa, tetapi berdasarkan pesan yang ingin disampaikan melalui desain. Selain itu, persepsi pengguna terhadap sebuah bahasa, kata, dan simbol harus dipertimbangkan dalam penyusunan *ontology*.

2) Taxonomy

Taksonomi adalah susunan bagian yang dibuat untuk melengkapi tujuan yang sesuai dengan konteks. Struktur ini disusun untuk memastikan integritas di seluruh kanal desain. Komponen ini juga dibuat untuk menyusun tiap komponen dengan menggunakan hierarki sehingga dapat dipahami oleh pengguna dengan mudah.

3) Choreography

Koreografi merupakan aturan yang dibuat dalam interaksi setiap bagian. Komponen ini berperan dalam pembuatan struktur dan memastikan setiap *flow* berjalan dengan baik sehingga tidak mengganggu pengguna. Inti dari koreografi adalah peletakan makna dan struktur menjadi sebuah *flow* dengan konteks spesifik. Koreografi berperan penting dalam pembuatan desain yang menggunakan beberapa perangkat dan aplikasi.

2.2.3.4 *Wireframe*

Wireframe merupakan representasi visual sederhana setiap layar untuk membantu desainer mengidentifikasi informasi dan elemen interaktif. *Wireframe* juga dapat berperan sebagai *blueprint* dari aplikasi yang akan dibuat. Biasanya *wireframe* digambar secara simpel dan fokus kepada struktur sebuah layar *smartphone*. Elemen lain seperti tekstur, bayangan, dan volume tidak dimasukan pada tahap ini (Cuello dan Vittone, 2013). *Wireframe* dapat dibuat dengan beberapa cara antara lain:

1) **Kertas**

   Wireframe yang dibuat dengan kertas adalah *wireframe* yang paling mendasar dan mudah dibuat. Pada dasarnya, semua *wireframe* dimulai dari kertas dan dikembangkan melalui *tools* lain seperti aplikasi desain.

2) ***Stencil***

   Cara berikutnya adalah dengan menggunakan *stencil* yang dapat memberikan kesan lebih formal, bersih, dan struktur yang teratur.

3) ***Template***

   Wireframe ini dibuat dengan menggunakan *file* digital untuk dibuat dengan menggunakan program desain seperti Photoshop, Illustrator, dan lainnya.

2.2.3.5 *Prototype*

Prototype merupakan representasi visual untuk dites secara internal untuk melihat kesalahan desain. Biasanya *prototype* dibuat dengan interaksi yang masih minimal. *Prototype* dapat dibuat berdasarkan *wireframe* yang sudah dibuat sebelumnya (Cuello dan Vittone, 2013).

Dalam pembuatannya, *prototype* dapat dibuat secara tradisional dengan gambar di kertas atau menggunakan program

yang dibuat khusus untuk menyusun *prototype*. Pemilihan cara pembuatannya bergantung pada kedekatan dan kemiripan dengan hasil akhir yang diharapkan. Cara pembuatan *prototype* dapat diklasifikasikan sebagai berikut:

1) **Dokumen interaktif**

   Prototype dapat disajikan dengan penambahan interaksi sehingga pengguna dapat mengeksplor satu bagian dokumen ke dokumen lainnya. Interaksi ini ditambahkan untuk mensimulasikan interaksi pada aplikasi.

2) ***Web versions***

   Dengan format ini, desainer dapat menunjukkan interaksi dan animasi sehingga pengguna seperti menggunakan aplikasi sungguhan,

3) **Format lainnya**

   Alternatif lainnya adalah dengan menggunakan *software* tertentu untuk membuat *prototype* yang mirip dengan hasi akhir aplikasi.

2.2.4 Pola Interaksi Aplikasi

Setiap aplikasi memikili banyak cara untuk berinteraksi dengan berbagai elemen yang ada pada layar *smartphone*. Dengan mengetahui perbedaan dari setiap cara berinteraksi, desainer dapat memanfaatkan dan menggunakan interaksi yang paling sering ditemui untuk dapat memastikan pengguna dapat menggunakan aplikasi dengan mudah dan nyaman. Menurut Welie (dalam Cuello dan Vittone, 2013), pola interaksi adalah kumpulan solusi desain yang terbukti bekerja lebih dari sekali. Pola interaksi ini sebaiknya digunakan sebagai pacuan dalam mendesain.

2.2.1.1 Navigasi

Navigasi yang mudah dan konsisten merupakan komponen penting yang dibutuhkan dalam *user experience* aplikasi. Dengan navigasi pengguna dapat mengetahui cara berpindah di dalam

aplikasi (Cuello dan Vittone, 2013). Berikut adalah macam-macam navigasi yang ada pada aplikasi:

1) Tabs

Tabs adalah navigasi yang digunakan untuk menyaring konten atau berpindah antara layar berdasarkan *information architecture*. Dalam pembuatan *tab*, ada baiknya desainer menambahkan penanda untuk memberikan informasi mengenai *tab* yang sedang dipilih. Pada *smartphone* Andorid, *tab* biasanya terletak dibagian atas layar.

Gambar 2.7 Peletakan *Tab* pada S*martphone*
Sumber: Cuello dan Vittone (2013)

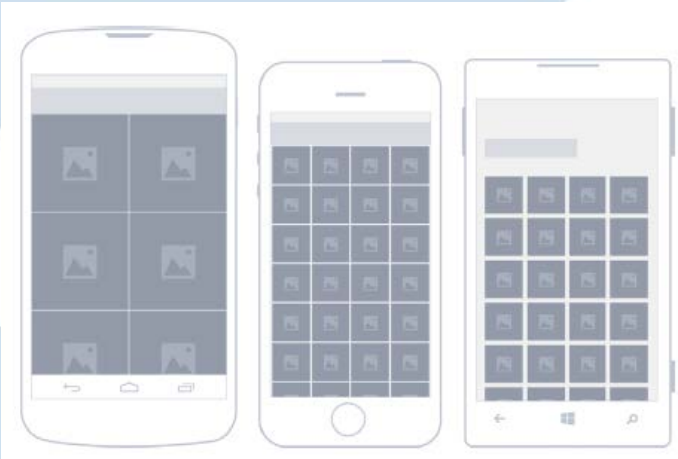
2) Lists

List merupakan susunan konten secara vertikal yang ditampilkan pada aplikasi. Navigasi ini digunakan untuk menunjukkan konten informasi dan pengguna dapat memilih salah satunya untuk mendapatkan informasi lanjutan. *List* dapat ditunjukan dengan gambar, tulisan, atau gabungan keduanya yang disusun dengan hierarki yang baik.

Gambar 2.8 Contoh Tampilan *List*
Sumber: Cuello dan Vittone (2013)

3) *Image gallery*

Navigasi ini adalah penyusunan gambar dengan menggunakan *grid*. Ketika gambar melebihi ruang yang diberikan, biasanya gambar akan di*crop* sesuai dengan *space* yang disediakan.



Gambar 2.9 Contoh Tampilan *Image Gallery*
Sumber: Cuello dan Vittone (2013)

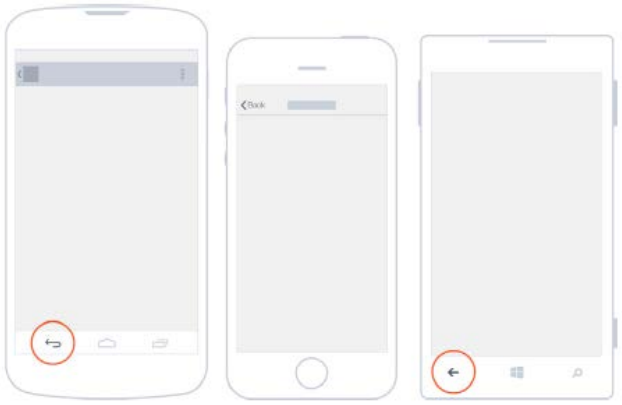
4) *Drawer menu*

Navigasi ini digunakan oleh pengguna untuk berpindah antar layar dengan cepat. Tampilan daftar konten yang tersembunyi dapat ditunjukkan dari sebuah tombol yang ditekan. Selain dengan tombol yang ditekan, menu ini dapat ditampilkan dengan menggunakan gestur menggeser layar.

Gambar 2.10 Contoh Tampilan *Drawer Menu*
Sumber: Cuello dan Vittone (2013)

5) Back

Navigasi ini digunakan pengguna untuk kembali ke layar yang dikunjungi sebelumnya. Navigasi ini sangat penting dalam aplikasi *mobile* dimana pengguna akan melakukan banyak perpindahan layar.



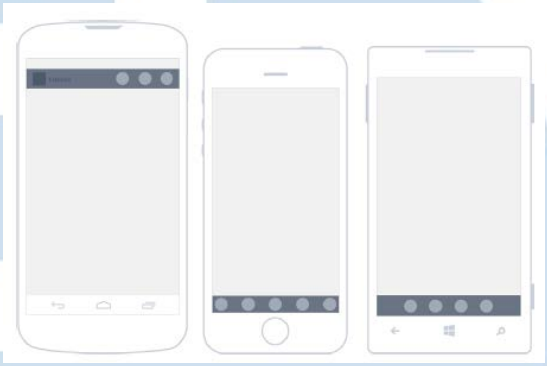
Gambar 2.11 Contoh Tampilan Navigasi *Back*
Sumber: Cuello dan Vittone (2013)

2.2.1.2 *Actions*

Pola interaksi berikutnya adalah *actions* atau aksi. Pengguna dapat melakukan beragam aksi dalam sebuah layar aplikasi untuk mengakses informasi yang disajikan. Letak aksi beragam bergantung dengan fungsi dan hierarki dari konten (Cuello dan Vittone, 2013).

1) *Action Bar*

Kumpulan dari beberapa aksi dapat dikelompokkan pada satu area dan membentuk *action bar*. Biasanya aksi yang dapat dilakukan di tampilkan dengan *icon*.



Gambar 2.12 Contoh Tampilan *Action Bar*
Sumber: Cuello dan Vittone (2013)

2) *Action Overflow*

Beberapa fitur tambahan pada aplikasi disembunyikan dan baru akan muncul ketika pengguna melakukan aksi terlebih dahulu. Informasi tambahan ini ditampilkan dalam format *list*.

Gambar 2.13 Contoh Tampilan *Action Overflow*
Sumber: Cuello dan Vittone (2013)

3) *Shortcuts*

Shortcut merupakan aksi yang disajikan supaya pengguna bisa menggunakan fitur aplikasi dengan cepat. Aksi ini juga dapat memudahkan pengguna ketika akan melakukan aksi yg berulang.

Gambar 2.14 Contoh Tampilan *Shortcuts*
Sumber: Cuello dan Vittone (2013)

4) *Search*

Aksi ini digunakan untuk mencaari konten atau informasi yang ada pada aplikasi. *Search* atau pencarian ini juga dapat menjadi sebuah fitur utama bagi aplikasi yang menyediakan banyak data. Pencarian ini dapat dilakukan dengan menggunakan teks maupun suara.

Gambar 2.15 Contoh Tampilan *Search*
Sumber: Cuello dan Vittone (2013)

5) *List Editing*

Aksi ini memberikan pengguna pilihan untuk mengatur elemen yang ada pada *list*. Cara kerja *list editing* ini dimulai dari pemilihan elemen dan kemudian pemilihan aksi yang akan dilakukan.

Gambar 2.16 Contoh Tampilan *List Editing*
Sumber: Cuello dan Vittone (2013)

2.2.1.3 *Dialogue Boxes*

Dialogue boxes atau kotak dialog biasanya digunakan untuk memberikan peringatan atau penjelasan singkat kepada pengguna sebelum pengguna melanjutkan aktivitas pada aplikasi (Cuello dan Vittone, 2013). Ketika kotak dialog ini sedang aktif, pengguna tidak dapat melakukan aktivitas lain di aplikasi. Kotak dialog biasanya disertai dengan opsi yang simpel seperti OK atau Batalkan.

Gambar 2.17 Contoh Tampilan *Dialogue Boxes*
Sumber: Cuello dan Vittone (2013)

2.2.1.4 *In-App Notifications*

Notifikasi dibutuhkan untuk memberikan konfirmasi aksi secara eksplisit kepada pengguna. Biasanya notifikasi ditunjukkan dengan pengumuman kecil yang muncul dalam beberapa saat. Berbeda dengan kotak dialog, notifikasi tidak memerlukan aksi dari

pengguna sehingga tidak mengganggu alur kerja pengguna (Cuello dan Vittone, 2013).

Gambar 2.18 Contoh Tampilan Notifikasi
Sumber: Cuello dan Vittone (2013)

Pada umumnya, notifikasi muncul dalam beberapa saat dengan teks singkat berisikan *feedback* untuk pengguna. Karena informasi yang disampaikan dengan notifikasi ini dapat terlewat oleh pengguna, informasi yang disajikan menggunakan pesan yang tidak kritis. Notifikasi dapat terletak dibagian atas maupun bawah layar aplikasi.

2.2.1.5 *Data Input*

Data input pada umumnya dilakukan dengan menggunakan *keyboard*. Selain memasukan teks dengan menggunakan *keyboard*, data juga dapat dimasukkan dengan menggunakan sensor lokasi, kamera, maupun mikrofon yang ada pada *smartphone* (Cuello dan Vittone, 2013).

2.2.1.6 *Gestures*

Pada aplikasi, segala aktivitas bergantung pada gerakan tangan pengguna untuk memanipulasi elemen yang ada di layar secara langsung (Cuello dan Vittone, 2013). Dengan adanya aktivitas ini, maka akan timbul interaksi antara pengguna dan aplikasi. Gestur sederhana seperti *tap, drag,* dan *swipe* yang hanya memerlukan gerakan dari satu atau dua jari sudah menjadi familiar pagi pengguna

dan paling sering digunakan. Dalam perancangan aplikasi, gestur dapat dimanfaatkan sebagai alur dalam navigasi konten. Berikut adalah beberapa gestur yang paling sering ditemui:

1) ***Tap***

   Gestur menyentuh layar dengan satu jari. Biasanya digunakan sebagai aksi utama.

2) ***Drag***

   Gestur dengan memindahkan jari pada permukaan layar tanpa melepaskan kontak dengan layar. Biasanya dipakai untuk berpindah layar.

3) ***Slide***

   Gestur menggeser layar dengan cepat tanpa melepaskan kontak dengan layar. Biasanya dipakai untuk berpindah layar atau *tab*.

4) ***Longpress***

   Gestur menyentuh layar dalam waktu tertentu tampa menggerakkan jari. Biasanya dipakai untuk masuk ke *edit mode*.

5) ***Double tap***

   Gestur menyentuh permukaan layar dua kali dengan cepat menggunakan ujung jari. Biasanya dipakai untuk *zoom in* atau *zoom out*.

6) ***Pinch and spread***

   Gestur menyentuh permukaan layar dengan dua jari dan menyebarkan atau mengecilkan jarak antara kedua jari. Biasanya dipakai untuk *zoom in* atau *zoom out*.

7) ***Rotate***

   Gestur menyentuh permukaan dengan dua jari sambil memutar pergelangan tangan. Biasanya dipakai untuk memutar gambar.

2.3 User Interface (UI)

Menurut Benyon (2019), *user interface* merupakan bagian dari sistem yang dapat disentuh atau dilihat oleh pengguna secara fisik, persepsi, maupun secara

konsep. Secara fisik, pengguna dapat secara langsung berinteraksi dengan sebuah perangkat dengan menyentuh tombol atau menggerakkan jari di atas layar dan pengguna akan mendapatkan *feedback*. *Interface* yang dirasakan dengan persepsi adalah tampilah layar yang bisa secara langsung dilihat, didengar, maupun dirasakan. Secara konsep, pengguna dapat berinteraksi dengan *interface* untuk mendapatkan informasi mengenai fungsi setiap konten yang ada pada desain.

2.3.1 Prinsip User Interface

Sebuah *interface* harus dapat merefleksikan kemampuan dari pengguna aplikasi dan dapat merespon setiap kebutuhan pengguna. *Interface* juga harus dapat dipelajari dengan mudah karena pengguna menggunakan *interface* untuk melakukan aktivitas bukan untuk mempelajarinya. Terdapat beberapa prinsip yang menjadi dasar dalam pembuatan *interface* yang efektif (Galitz, 2007).

1) ***Accessibility***

   Sebuah sistem harus dapat digunakan oleh berbagai macam orang tanpa desain khusus atau modifikasi.

2) ***Aesthetically Pleasing***

   Menyediakan visual yang dapat menarik perhatian. Desain yang ditampilkan juga harus dapat menyampaikan pesan secara cepat dan jelas. Prinsip ini dapat diterapkan dengan menggunakan kontras antara elemen, membuat pengelompokkan, membuat jajaran layar dan elemen, menggunakan warna dan grafis secara efektif, dan memberikan representasi secara tiga dimensi.

3) ***Availability***

   Sebuah *interface* harus dapat diakses dimanapun dan kapanpun.

4) ***Clarity***

   Interface harus dapat terlihat dengan jelas dalam segala aspek yang mencangkup elemen visual, fungsi, metafor, dan teks.

5) ***Compatibility***

Desain yang dibuat harus sesuai dengan kebutuhan pengguna, tugas yang dilakukan, dan produk yang akan ditawarkan.

6) ***Configurability***

Interface harus dapat memberikan kesempatan bagi pengguna untuk mengatur konfigurasi *interface* untuk dapat memberikan rasa kontrol dan mengajak pengguna menjadi aktif dalam penggunaan sebuah *interface.*

7) ***Consistency***

Keseluruhan sebuah sistem harus terlihat sama dan seragam. Komponen-komponen yang ada di dalam *interface* harus memiliki tampilan, penggunaan, dan bekerja serupa. Setiap aksi yang serupa harus memberikan *feedback* yang serupa juga.

8) ***Control***

Setiap kontrol yang ada pada sistem dikendalikan oleh pengguna. Setiap fungsi dan proses harus dibuat dengan perspektif pengguna dan disesuaikan dengan kemampuan, kebiasaan, dan preferensi pengguna.

9) ***Directness***

Pengguna harus dapat melakukan sebuah tugas secara langsung. Alternatif dari cara penggunaan dan efek dari aksi harus terlihat dengan jelas.

10) ***Efficiency***

Sistem harus dapat mengantisipasi kebutuhan pengguna kapanpun. *Interface* meminimalisasi pergerakan tangan dan mata.

11) ***Familiarity***

Konsep yang digunakan harus familiar dengan setiap pengguna. *Interface* yang dibuat harus natural dan didasari oleh kebiasaan pengguna.

12) ***Flexibility***

Setiap sistem harus dapat mengakomodasi perbedaan kebutuhan dari setiap pengguna dari segi pengetahuan, pengalaman, preferensi, kebiasaan, dan kondisi.

13) ***Forgiveness***

Mencegah *error* supaya tidak terjadi kapan saja. *Error* yang mungkin muncul adalah kesalahan yang dapat dimaklumi.

14) ***Immersion***

Interface harus dapat membuat setiap pengguna terimersi ke dalam setiap tugas.

15) ***Obviousness***

Sistem yang dibuat harus dapat dipahami dan dimengerti dengan mudah. Setiap alur dalam *interface* harus disusun berurutan sesuai dengan urutan konteks yang diberikan.

16) ***Operability***

Setiap sistem harus dapat digunakan oleh semua orang tanpa memandang kemampuan fisik.

17) ***Perceptibility***

Setiap desain harus dapat dirasakan oleh semua orang tanpa memandang kemampuan indera.

18) ***Positive First Impression***

Impresi pertama dari pengguna dapat mempengaruhi persepsi dan sikap dalam interaksi. Impresi ini dapat terbuat dari pengguna pertama kali memasuki sistem.

19) ***Predictability***

Pengguna harus dapat mengantisipasi alur perjalanan tugas yang dikerjakan. Setiap ekspektasi harus dapat dipenuhi secara seragam dan sepenuhnya.

20) ***Recovery***

Sebuah sistem harus dapat memberikan aksi untuk mengembalikan atau membatalkan sebuah aksi. Sistem juga

harus memastikan pengguna tidak akan kehilangan progres yang telah dibuat.

21) ***Responsiveness***

Permintaan dari pengguna harus dapat direspon oleh sistem secepat mungkin. Sistem juga harus memberikan informasi langsung dengan menggunakan visual, teks, atau audio.

22) ***Safety***

Melindungi pengguna untuk tidak melakukan kesalahan dengan menyediakan tanda visual, peringatan, pilihan, dan lainnya.

23) ***Simplicity***

Interface yang diberikan harus diberikan harus sesederhana mungkin. Hal ini dapat dicapai dengan hierarki, membuat aksi simpel, menghilangkan elemen yang tidak penting, dan lainnya.

24) ***Transparency***

Membuat pengguna fokus dalam pengerjaan tugas tanpa memikirkan mekanik dari *interface*.

25) ***Trade-offs***

Desain akhir didasari oleh penyesuaian keseimbangan walaupun bertabrakan dengan prinsip desain. Kebutuhan pengguna lebih dipentingkan daripada kebutuhan teknis.

26) ***Visibility***

Setiap status dan metode yang disediakan harus terlihat dengan jelas. *Visibility* dapat ditingkatkan dengan menggunakan hierarki dan *context sensitivity*.

2.4 User Experience (UX)

Menurut Deacon (2020), *user experience* adalah penyusunan sebuah produk yang memberikan solusi bagi masalah *customer* untuk mendapatkan kesetiaan dan relasi bisnis dalam jangka panjang. *User experience* dibuat untuk dapat memberikan pengalaman yang berguna dan relevan bagi pengguna. UX menyangkup desain keseluruhan proses penyatuan produk, desain, *usability,* dan

fungsi. UX juga dapat dilihat sebagai cara untuk meningkatkan kualitas interaksi antara pengguna dan aspek dari sebuah perusahaan.

2.4.1 Prinsip User Experience

Prinsip dari *user experience* dapat diartikan sebagai proses untuk mengarahkan produk menjadi sebuah desain berdasarkan elemen yang dipilih. Prinsip desain ini juga dapat dilihat sebagai inti dari desain *user experience* itu sendiri (Deacon, 2020). Prinsip-prinsip tersebut terdiri dari:

1) ***Meeting the user's needs***

   Prinsip ini merupakan tujuan utama dari pembuatan *user experience*, yaitu memenuhi kebutuhan dari pengguna. Desainer harus dapat mempelajari kebutuhan dari setiap pengguna untuk dapat meningkatkan *user experience*.

2) ***Know your current stage in the designing process***

   Dalam perancangan *user experience*, ada pentingnya bagi desainer untuk selalu paham tahapan yang sedang dilakukan untuk dapat mengetahui survei dan data apa yang diperlukan untuk melakukan riset kepada pengguna.

3) ***A well-defined hierarchy***

   Hierarki pada *user experience* berfungsi untuk memastikan pengguna dapat menavigasi tanpa stres. Hierarki terbagi menjadi dua yaitu hierarki susunan konten pada seluruh proses desain dan hierarki yang membuat pengguna dapat menavigasi sebuah halaman atau sebuah bagian tanpa halangan.

4) ***Consistency***

   Penggunaan desain yang konsisten dan serupa dengan produk yang sudah ada dapat membuat pengguna terbiasa dalam penggunaan dan membutuhkan waktu singkat untuk membiasakan diri.

5) ***Accessibility and usability***

Desainer harus memastikan desain yang dibuat dapat dipkai dan diakses oleh banyak pengguna. Setiap desain juga harus dapat digunakan dengan mudah.

6) ***Simple metaphor***

User experience yang baik harus menghindari istilah-istilah ambigu dan menggunakan kata-kata yang mudah dipahami.

2.4.2 Pola Visual Mata

Menurut Pernice (2019), pengguna tidak membaca setiap kata yang ada pada halaman melainkan melakukan pemindaian terhadap sebuah tampilan. Hal ini dilakukan untuk meningkatkan efisiensi dalam mendapatkan informasi dalam waktu dan upaya minimal. Terdapat empat pola yang biasanya dilakukan untuk memindai informasi yaitu:

1) ***F-Pattern***

Pola ini merupakan pola yang dinamai berdasarkan pola gerakan mata yang menyerupai huruf F. Pengguna biasanya terpaku pada huruf yang ada pada atas halaman. Dalam bahasa kiri-ke-kanan, tulisan yang berada pada sisi kiri atas halaman memiliki tingkat keterbacaan lebih tinggi dibandingkan dengan tulisan yang ada pada bagian kanan bawah.

Gambar 2.19 *F-Pattern*
Sumber: Nielsen dan Pernice (2010)

2) *Spotted Pattern*

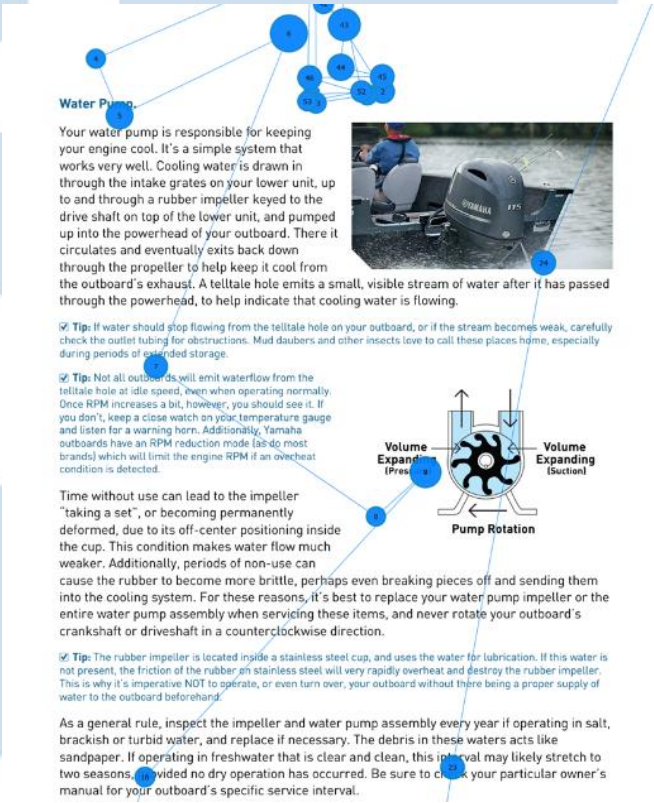
Pola ini merupakan pola pemindaian yang berpusat pada kata-kata tertentu yang tersebar pada suatu halaman. Pola ini akan lebih praktis daripada *F-Pattern* apabila desainer dapat memberikan penamaan poin yang baik sehingga kata penting yang disajikan terlihat beda daripada teks lainnya dan membentuk daftar poin. Kata penting dapat disajikan dengan berikut:

- Tampilan kata dibuat lebih menonjol daripada informasi lainnya karena adanya perbedaan dalam tampilan. Perbedaan dapat terlihat dari warna, ukuran, ketebalan tulisan, daftar poin, dan lainnya.
- Menggunakan kata yang menyerupai tujuan yang diinginkan oleh pengguna. Contohnya adalah penggunaan huruf kapital untuk alamat, informasi dalam bentuk angka, dan lainnya.

Gambar 2. 20 Pola Visual *Spotted Pattern*
Sumber: Pernice (2019)

3) *Layer-cake Pattern*

Pola ini merupakan pola paling efektif dalam pemindaian halaman Syang terdiri dari fokus pada judul dan sub judul di halaman. Penyusunan halaman pada pola ini terdiri dari judul, sub judul, dan teks yang ada diantara keduanya. Pengguna dapat memindai judul yang ada pada halaman hingga pengguna menemukan judul yang diminati dan kemudian mengamati teks yang ada di bawahnya. Pola ini biasanya terlihat seperti tumpukan garis horizontal dengan spasi kosong diantaranya sehingga menyerupai lapisan kue.



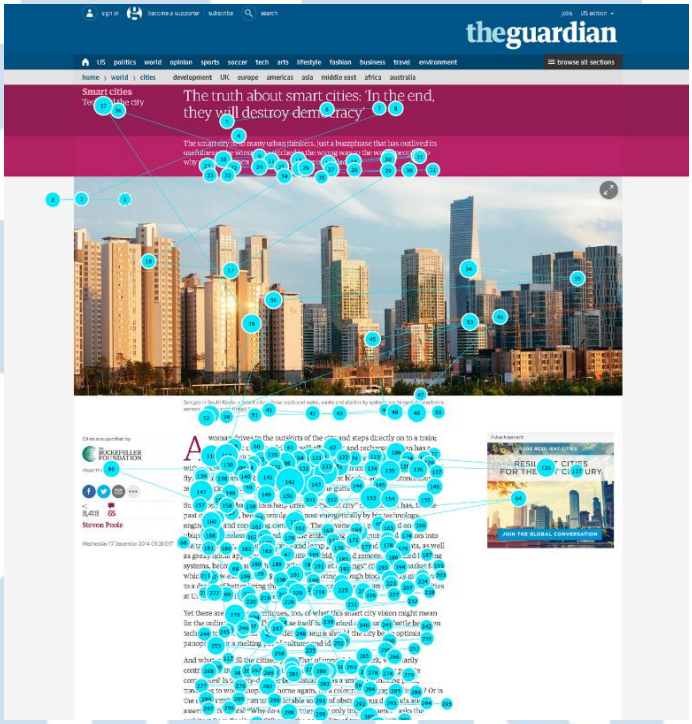
Gambar 2.21 Layer-cake Pattern
Sumber: Pernice (2019)

4) *Commitment Pattern*

Pola ini menunjukkan pola pembacaan secara tradisional bukan memindai. Pengguna dengan pola ini fokus kepada hampir seluruh konten yang ada pada paragraf teks. Pola ini muncul ketika pengguna sangat tertarik atau bermotivasi untuk

membaca konten yang disajikan. Pola ini memberikan pengertian yang sangat tinggi tetapi membutuhkan waktu yang lama. Motivasi pengguna dapat muncul apabila:

- Pengguna mengetahui dan percaya dengan sumber informasi.
- Pengguna memiliki loyalitas terhadap suatu *brand.*
- Pengguna percaya bahwa halaman yang dikunjungi adalah tempat terbaik untuk mendapatkan informasi.



Gambar 2.22 *Commitment Pattern*
Sumber: Pernice (2019)

2.5 BPJS Kesehatan

Badan Penyelenggara Jaminan Sosial (BPJS) merupakan badan hukum publik yang dibuat untuk dapat menyelenggarakan program jaminan sosial. BPJS terbagi menjadi dua yaitu BPJS Kesehatan dan BPJS Ketenagakerjaan. Sesuai dengan namanya, BPJS Kesehatan merupakan badan hukum yang menyelenggarakan program jaminan kesehatan. Jaminan kesehatan adalah jaminan yang melindungi kesehatan setiap peserta agar peserta bisa mendapatkan

manfaat dan perlindungan pemeliharaan kesehatan yang menjadi kebutuhan dasar kesehatan. BPJS Kesehatan mulai berjalan pada tanggal 1 Januari 2014.

Peserta BPJS Kesehatan adalah seluruh penduduk Indonesia dan juga orang asing yang sudah bekerja minimal enam bulan di Indonesia dan sudah membayar iuran. Peserta BPJS Kesehatan dibagi menjadi dua kelompok, yaitu Penerima Bantuan Iuran (PBI) jaminan kesehatan dan bukan Penerima Bantuan Iuran (PBI). PBI merupakan peserta Jaminan Kesehatan untuk fakir miskin dan tidak mampu sesuai dengan amanat UU SJSN dan iuran peserta dibayarkan oleh pemerintah sebagai peserta program Jaminan Kesehatan (Kementrian Kesehatan, n. d).

2.5.1 JKN

Menurut Kementrian Kesehatan (n. d), Jaminan Kesehatan Nasional (JKN) adalah bagian dari Sistem Jaminan Sosial Nasional (SJSN) yang diselenggarakan berdasarkan mekanisme Asuransi Kesehatan Sosial yang wajib berdasarkan peraturan Undang Undang no. 40 Tahun 2004 tentang Sistem Jaminan Sosial Nasional. Tujuan dibuatnya JKN ini adalah untuk melindungi seluruh penduduk Indonesia dalam sistem asuransi sehingga penduduk Indonesia bisa memenuhi seluruh kebutuhan dasar kesehatan masayarakat. JKN memiliki beberapa prinsip yang mengacu pada prinsip sistem Sistem Jaminan Sosial Nasional (SJSN), yaitu:

1) **Prinsip Kegotongroyongan**

   Prinsip ini diartikan dengan peserta yang dapat saling membantu peserta yang kurang mampu, sakit, dan berisiko tinggi.

2) **Prinsip Nirlaba**

   Tujuan dibentuknya JKN bukanlah untuk mencari laba melainkan untuk memenuhi kepentingan yang dimiliki oleh setiap peserta.

3) **Prinsip Portabilitas**

   Jaminan yang diberikan bersifat berkelanjutan meskipun peserta berpindah tempat tinggal atau pekerjaan dala wilayah Indonesia.

4) **Prinsip Kepesertaan Bersifat Wajib**

Dibuat sehingga setiap rakyat Indonesia menjadi peserta yang dapat dilindungi.

5) **Prinsip Dana Amanat**

Setiap dana yang dikumpulkan adalah dana titipan kepada penyelenggara untuk dapat dikelola untuk mengoptimalkan kesejahteraan peserta.

6) **Prinsip Hasil Pengelolaan Dana Jaminan Sosial**

Prinsip ini digunakan untuk mengembangkan program dan digunakan untuk kepentingan setiap peserta.

2.5.2 Kanal Layanan BPJS Kesehatan

Peserta layanan BPJS Kesehatan dapat melihat informasi, memberi pangaduan, memberi kritik dan saran, melakukan pendaftaran dan pengubahan data peserta melalui beberapa media yang telah disediakan. Media inilah yang disebut dengan kanal layanan BPJS Kesehatan (BPJS Kesehatan, 2020). Kanal layanan tersebut terdiri dari:

1) Aplikasi Mobile JKN
2) BPJS Kesehatan *Care Center* 1500 400
3) *Mobile Customer Service* (MCS)
4) *Website* BPJS Kesehatan
5) Mall pelayanan publik
6) Kantor cabang dan kantor kabupaten/kota
7) Pendaftaran badan usaha integrasi dengan OSS (Online Single Submission)
8) Portal bersama BPJS Kesehatan dan BPJS Tenaga Kerja

2.5.2.1 Aplikasi Mobile JKN

Mobile JKN merupakan media aplikasi yang dibuat untuk dapat mempermudah peserta JKN-KIS atau calon JKN-KIS dalam mengakses layanan yang diberikan (BPJS Kesehatan, 2020).

Aplikasi ini dirilis di Play Store pada tanggal 17 April 2016. Aplikasi ini memiliki berbagai fitur yang dapat digunakan oleh pengguna antara lain:

1) **Fitur Pendaftaran Peserta**
   Pengguna aplikasi dapat mendaftarkan diri sebagai peserta PBPU dengan menggunakan nomor KTP.
2) **Fitur Peserta**
   Pengguna dapat melihat informasi diri sendiri dan keluarga.
3) **Fitur Ubah Data Peserta**
   Pengguna dapat mengubah data peserta seperti nomor *handphone, email,* alamat, dan pindah kelas.
4) **Fitur Ketersediaan Tempat Tidur**
   Pengguna dapat melihat informasi terkait ketersediaan tempat tidur di rumah sakit sesuai dengan kelas.
5) **Fitur Obat yang Ditanggung**
   Pengguna dapat melihat jenis obat yang dapat ditanggung.
6) **Fitur Premi**
   Pengguna dapat melihat tagihan peserta dan anggota keluarganya.
7) **Fitur Jadwal Tindakan Operasi**
   Pengguna dapat melihat jadwal operasi yang akan dilakukan dan dapat melihat jadwal operasi yang tersedia di rumah sakit.
8) **Fitur Pendaftaran Pelayanan**
   Menampilkan pendaftaran FKTP pengguna dan pendaftaran pada Fasilitas Kesehatan Rujukan Tingkat Lanjutan dan pengguna dapat memberikan penilaian serta masukan pelayanan yang pernah diterima.
9) **Fitur Pendaftaran *Auto Debit***
   Pengguna dapat melihat tata cara pendaftaran auto debit.

10) **Fitur Pembayaran**

Pengguna dapat melihat tata cara pembayaran dengan bank dan *E-money*.

11) **Fitur Catatan Pembayaran**

Pengguna dapat melihat riwayat pembayaran iuran dan denda yang pernah dilakukan.

12) **Fitur Riwayat Pelayanan**

Pengguna dapat melihat riwayat dan informasi pelayanan yang pernah dilakukan sebelumnya.

13) **Fitur Skrining Riwayat Kesehatan**

Penulis dapat mengisi pertanyaan terkait riwayat kesehatan.

14) **Fitur Cek *Virtual Account***

Pengguna dapat melihat nomor *virtual account* peserta.

15) **Fitur Info JKN**

Pengguna dapat melihat informasi mengenai program JKN-KIS seperti manfaat, persyaratan, hak, kewajiban, dan sanksi.

16) **Fitur Lokasi**

Pengguna dapat melihat posisi kantor BPJS Kesehatan dan Fasilitas Kesehatan tersekat.

17) **Fitur Pengaduan Keluhan**

Pengguna dapat meemberikan pengaduan secara telepon maupun tertulis melalui aplikasi.

UNIVERSITAS MULTIMEDIA NUSANTARA