

## BAB II

### TINJAUAN PUSTAKA

#### 2.1. *Mobile Application*

Berdasarkan kamus *Oxford dictionaries*, arti dari kata *Mobile* adalah terhubung dengan ponsel atau telepon genggam, tablet dan internet. Dalam jurnal *Mobile Application and Its Global Impact* (2010) menjelaskan bahwa *mobile application* berjalan pada perangkat seluler yang dapat dibawa, mudah digunakan dan dapat diakses dari dimanapun yang banyak digunakan untuk berhubungan atau mengontak teman, menelusuri internet, mengatur *file* konten, membuat suatu dokumen, juga hiburan.

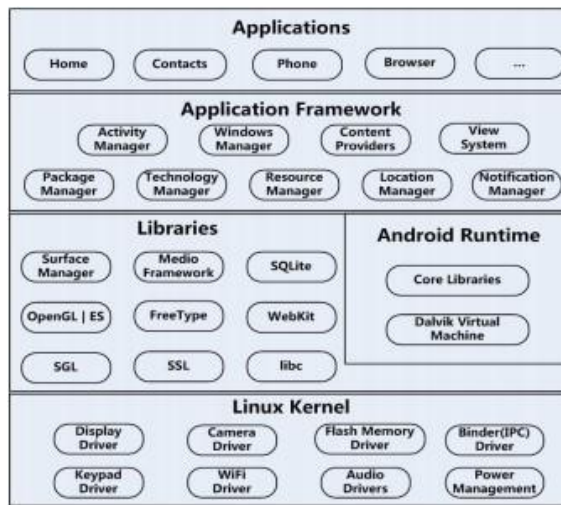
*Mobile application* terdiri dari *software/set* program yang dijalankan pada perangkat seluler dan menjalankan *task* dari *user* dan dari sisi teknikal, beda *mobile application* dijalankan pada platform yang berbeda seperti *iPhone*, *Blackberry*, *Android*, *Symbian*, *Windows* dan mesin virtual seperti *Java/J2ME*, *BREW*, *Flash Light*, *Silverlight*. *Mobile Application* dibagi menjadi 6 macam kategori yaitu sebagai berikut.

- A. *Communications* : penelusuran *internet*, *email* klien IM, dan *networking* sosial.
- B. *Games* : *puzzle* (strategi), *cards* (*casino*), dan *Action* (Petualangan).

- C. *Multimedia* : grafik (*image viewer*), penampil presentasi, *video*, dan *audio player*.
- D. *Productivity* : kalender, kalkulator, diari (buku harian), dan catatan (*notepad, memo, word processors, spreadsheets*).
- E. *Travel* : petunjuk kota, konverter mata uang, penerjemah, GPS/peta, rencana perjalanan/jadwal perjalanan, dan cuaca.
- F. *Utilities* : *profile manager, screensaver*, catatan alamat, *task manager, call manager, file manager* (hal.104).

### **2.1.1. Platform Android**

Berdasarkan jurnal *Research on Development of Android Applications*, Sasankar dan Kosarkar (2016) mengatakan bahwa *Android* adalah sebuah *system operation* sumber terbuka yang berdasarkan *Linux kernel* dan diluncurkan oleh *Google*. *Operating system* perangkat seluler terbatas dengan *hardware, storage space, power dissipation* dan *mobility conditions*. *Android* adalah *operating environment* yang luas dan berdasarkan *Linux® V2.6 kernel*, dan juga merupakan sistem yang berlayar. Berikut merupakan *architecture* dari sistem *Android* (hal. 28).



Gambar 2.1. Contoh *Brand* Dengan Logo *Wordmakrs*  
 (<https://brand.netflix.com/en/assets/>, 2020)

Berdasarkan buku *Android™ Application Development For Dummies® 3rd Edition*, Burton (2015) menjelaskan bahwa terdapat banyak versi *Android* yaitu nama versi dan kode versi, setiap versi memiliki kode versi yang berbeda. Berikut merupakan tabel nama versi dengan kode versi *Android* yang sesuai dan dilengkapi dengan data berdasarkan *website* resmi *Android* untuk versi yang terbaru lainnya (hal. 48).

Tabel 2.1. Tabel Kode Versi dan Nama Versi *Platform Android*

No.	Versi <i>Platform</i>	Nama Kode
1	4.0 - 4.0.3	<i>Ice Cream Sandwich</i>
2	4.1 - 4.3	<i>Jelly Bean</i>
3	4.4	<i>Kit Kat</i>
4	4.4W	<i>Wearables</i>
5	5.0 - 5.1	<i>Lollipop</i>
6	6.0	<i>Marshmallow</i>
7	7.0 - 7.1	<i>Nougat</i>
8	8.0.0 - 8.1.0	<i>Oreo</i>
9	9	<i>Pie</i>
10	10	<i>Android 10</i>
11	11	<i>Android 11</i>

## **2.2. User Experience**

Dalam Aplikasi terdapat UI UX, UX merupakan singkatan dari kata *user experience* atau *User Experience Design* (abbreviation UX, UXD). Pada buku *UX Design For Startups*, Treder (2013) mengatakan bahwa *user experience* adalah sebuah fokus disiplin pada mendesain *experience* dari sebuah produk dari ujung ke ujung, mendesain suatu *experience* adalah untuk merencanakan dan melaksanakan sejumlah aksi yang diharapkan dapat menghasilkan perubahan dari perilaku *target group* ketika melakukan interaksi dengan produk (hal. 19).

### **2.2.1. User Experience Design**

Getto dan Cao (2016) menjelaskan bahwa terdapat tahapan dalam merancang *user experience*. Tahapan itu merupakan *preliminary research*, *maintenance*, *prototyping* dan *user testing*. Berikut merupakan penjelasan tahapan (hal. 26).

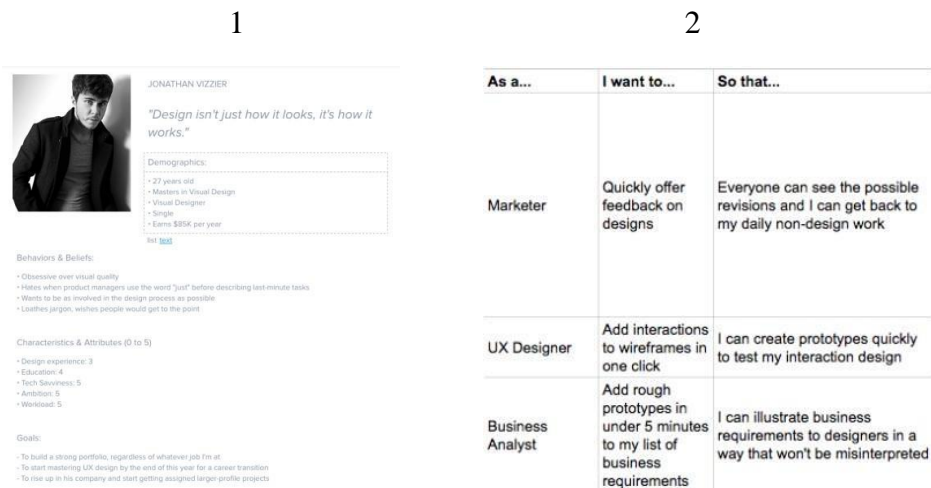
#### **2.2.1.1. Preliminary research**

*Preliminary* merupakan langkah yang penting dalam merancang *user experience* namun sering dilewati, pada langkah ini merancang sebuah proyek rencana yang berartikulasi pada bagaimana cara bekerja *stakeholders* dengan proyek yang dirancang. Riset dimulai dari *contextual inquiry*, mempelajari kultur individu untuk menanyakan apa kebutuhan dan keperluannya, dari data itu diperlukan sebagai dasar pembuatan proyek (hal. 28).

Ada 2 macam proyek yang dapat dibuat yaitu *hypotical projects* dan *client-based projects*. *Hypotical projects* merupakan proyek dimana

perancang menggunakan produk yang sudah ada dan didesain ulang atau dijadikan produk hipotetis sendiri. perancang juga dapat menggunakan dua produk yang berbeda lalu dijadikan satu. *Client-based projects* merupakan jenis proyek dimana perancang merancang suatu produk sesuai dengan permintaan klien untuk mendesain produk baru atau juga bisa mendesain ulang untuknya (hal.17).

Hal yang perlu disediakan pada tahapan ini adalah *personas*, *user stories*, *user scenarios*, *user flows*. *Personas* merupakan profile *user* buatan yang mewakili *user* asli sebagai kunci demografi dari *user research* (gambar 1 pada gambar 2.2.). *User stories* mewakili *user action* dan motivasi yang penting, ini dapat dibuat pada *Google spreadsheet* dan merupakan cara tercepat (gambar 2 pada gambar 2.2.).



Gambar 2.2. *Personas*.(1), *User Story* (2)  
(*UX Design The Definitive Beginners Guide*, 2016)

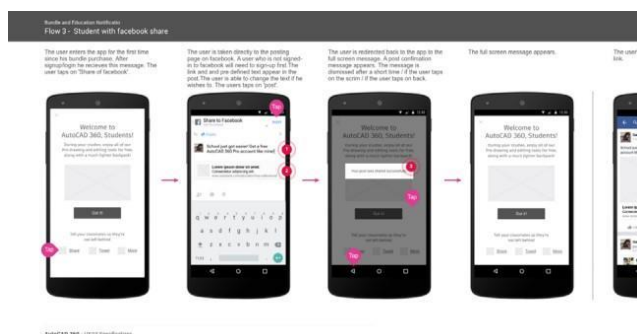
*User scenarios* menjelaskan situasi pada *user story*, dapat membantu perancang untuk berempati dengan *user story* karena terdapat detail seorang

individu. Untuk mendapatkan *user scenario* dengan cepat, perlu menambahkan kolom tambahan pada *spreadsheet user story* dimana perancang dapat menyertakan *scenario* (hal. 32).

As a...	I want to...	So that...	Scenario 1	Scenario 2	Scenario 3
Marketer	Quickly offer feedback on designs	Everyone can see the possible revisions and I can get back to my daily non-design work	It's 7:30PM on a Friday night. John should be home already, but he's staying late wrapping up the copy for a new landing page and to go live next week. He sees an email from the designer on another project asking for some emergency copy since they just realized the header and first paragraph is still in Lorem Ipsum. He feels frustrated because he asked the designers to insert some rough copy as a starting point. John's already clocked in 50 hours for the week, so he wants a smooth way to give his feedback as easily and quickly as possible so he can head home.		

Gambar 2.3. *User Scenario*  
(UX Design The Definitive Beginners Guide, 2016)

*User flows* merupakan sketsa dari jalan *user* saat menggunakan produk yang dirancang dan dapat menjadi sebagai peta atau gambaran garis yang menggambarkan bagaimana *user* navigasi pada produk. Apabila digabungkan dengan *analysis task* akan gambaran yang jelas dalam merancang jalan termudah untuk *user* dapat sampai ke *goal* (hal. 33).



Gambar 2.4. *User Flow*  
(UX Design The Definitive Beginners Guide, 2016)

### 2.2.1.1. Maintenance

Dalam maintenance *user experience*, dibutuhkanannya suatu *information architecture* sebagai sebuah dasar dari struktur informasi dan *flows*

berjalannya produk. Sebelum mendesain produk perlu membuat *information architecture* produk dahulu agar dapat dicatat dan dikembangkan selama menjalani proses desain.

Hal yang perlu disediakan pada tahapan ini adalah *sitemaps*, *taxonomies*, dan *content governance plans*. *Sitemaps* menggambarkan semua *page* dari produk dan bagaimana mereka saling berhubungan, biasanya dalam diagram sederhana dari gambaran garis (hal. 36).



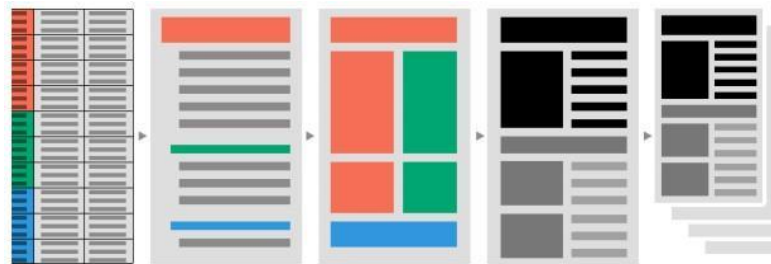
Gambar 2.5. *Sitemaps*  
(*UX Design The Definitive Beginners Guide*, 2016)

*Taxonomies* adalah daftar kata-kata semua informasi pada suatu produk yang membantu dalam memastikan informasi dalam sebuah produk yang dapat digunakan, yang sah, dan dapat dikontrol. *Content governance plans* merupakan sebuah strategi untuk *updating*, *revisiting*, dan *maintaining* semua konten dalam produk. Biasanya termasuk beberapa sub-elemen yang akan membantu klien dalam mempertahankan produknya (hal. 37).

### 2.2.1.3. *Prototyping*

Setelah selesai proses riset *user* dan kebutuhannya serta struktur konten, lalu mulai merancang desain produk. Dalam desain *user experience* dibutuhkan

sebuah aplikasi *prototyping*, kunci dalam mendesain *user experience* adalah mengetahui membuat apa dan kapan membuat. Perlu untuk memikirkan hubungan antara *prototype*, *user interaction*, dan *UI elements* (hal.38).



Gambar 2.6. Proses *Prototyping*, *Interaction design*, dan *Visual design* (*UX Design The Definitive Beginners Guide*, 2016)

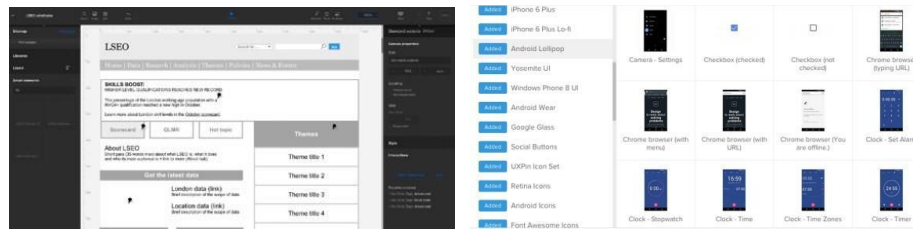
Hal yang perlu disediakan pada tahapan ini adalah *wireframes*, *UI elements*, dan *mockups*. Berikut merupakan penjelasan hal-hal yang disebut. *Wireframe* merupakan alat yang sangat berguna dalam mengembangkan visi yang disatukan pada produk yang lebih fokus pada struktur konten dibandingkan bagusnya *visual* (gambar 1). Apabila *wireframe* dapat diklik, maka termasuk sebagai *wireframe low-fidelity* interaktif (hal.39).

*UI elements* merupakan komponen atomik dari produk yang akan terlihat oleh *user* (gambar 2). Diperlukan untuk memasukkan beberapa *UI elements* dalam *prototype*, seperti *buttons*, *links* dan *spaces* yang dibatasi dari *space* lainnya (seperti *headers*, *footers*, *scrollbars*, *images*, *pages*, dan lainnya), karena diperlukan *user* untuk dapat berinteraksi dengan *prototype* yang dirancang. Semakin banyak *fidelity* akan semakin banyak pula *UI elements* (hal. 40).



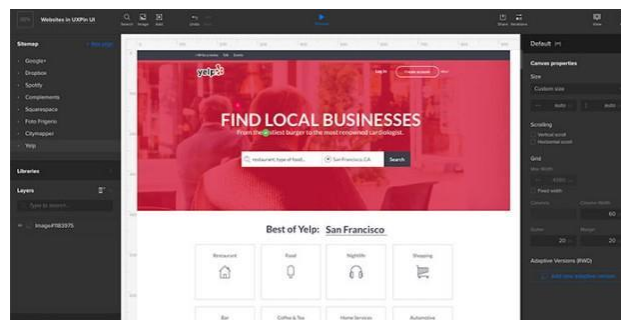
1

2



Gambar 2.6. Proses *Wireframe* (1), Proses *UI Elements* (2)  
(*UX Design The Definitive Beginners Guide*, 2016)

*Mockups* merupakan penggambaran produk yang lebih lengkap, biasanya merupakan *medium* sampai *high-fidelity*. Bentuk bisa berupa statis atau dapat diklik tergantung pada *tool* apa yang digunakan untuk mengembangkan *Mockup*.



Gambar 2.7. Proses *Mockup*  
(*UX Design The Definitive Beginners Guide*, 2016)

#### 2.2.1.4. *User testing*

Proses *user experience* hampir tidak pernah berjalan secara linear, terkadang harus menggabungkan tahapan karena alasan batasan waktu atau pertimbangan lainnya. Terkadang sebagai perancang tidak memiliki banyak waktu, maka biasanya digabungkan *user research* dengan testing

*preliminary usability*. Kemudian, melakukan *testing* cepat dengan *prototype fidelity* yang lebih tinggi.

	User 1	User 2	User 3	User 4	User 5
User feels interface is overwhelming	Red				
Prefers "search" over browsing the categories	Red	Orange			Blue
Requested that "Accepts Credit Cards" be a top-level filter	Red		Green		
Wants photo gallery accessible on results page to assess restaurant ambience					
Bookmark feature was frustrating		Orange		Yellow	
Needs clearer indication of price ranges	Red				Blue
Felt it was easy to sort restaurants by "Open Now"			Green		
Could not find the Events tab				Yellow	

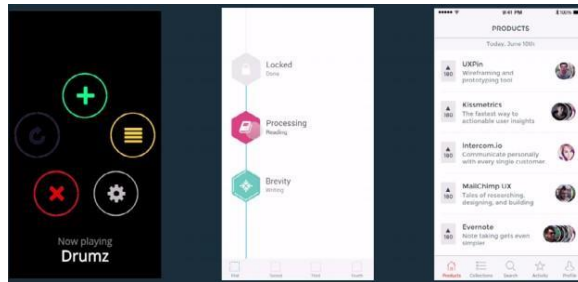
Gambar 2.8. *User Research and User Testing*  
(*UX Design The Definitive Beginners Guide*, 2016)

*User testing* didesain untuk *test* kegunaan dan berjalannya pada sebuah aplikasi, seberapa efektif dalam membantu *user* mencapai *goal*. *Goal user* dapat didesain *script usability testing* dengan mudah melalui tahapan tertentu.

Sebelum *testing*-- *Usability test plan* menyimpulkan latar belakang, *goals*, *objectives*, dan metodologi untuk *testing*. Hal seperti menentukan antara *usability test on-site* dan *remote*, menentukan antara dimoderasi dan tidak, dan berapa banyak orang yang *testing*. *User testing script* diperlukan membuat *script* untuk seluruh *experience* untuk menjalankan *user test* efektif. *Test user* harus versi dari *persona*.

Setelah *testing Usability reports*, dalam membuat *usability report* diperlukan sebuah pengantar mengenai apa yang efektif dari *prototype* yang dirancang dan juga apa yang tidak. *Journey map* memiliki kemiripan dengan *user flow*, namun lebih detail. Dengan data *user testing* akan lebih menjelaskan bagaimana *user* akan navigasi hasil akhir produk. *Refined prototype*, tujuan utama dalam *user testing* adalah untuk mengembangkan

*prototype* yang dirancang. Diperlukan untuk *test* masing-masing iterasi *prototype* pada para *user* (hal 46).



Gambar 2.9. *User Testing*  
(*UX Design The Definitive Beginners Guide*, 2016)

### 2.2.2. *Gamification*

Pada buku *Digital Health and the Gamification of Life*, Mauro dan Moretti (2018) mengatakan bahwa *Gamification* didefinisi sebagai penggunaan elemen *game* pada konteks *non-game*, ini merupakan strategi yang bagus dalam banyak aspek di kehidupan manusia. Melalui *game* atau elemen yang dapat dimainkan dapat memperkuat motivasi pada individu, mempermudah dalam menjalankan aktivitas yang berat dan sulit (hal. 1).

Cao, Sergio dan Ellis menjelaskan bahwa *gamification* tidak hanya untuk *entertainment (game)*, terdapat juga pada aplikasi atau situs lainnya untuk menyebarkan *awareness* suatu *brand*. *Habit loop* merupakan hal penting yang membuat *video game* menyenangkan, hal ini dapat memandu bahkan melatih para *user* untuk melakukan *action* yang disukai berulang-ulang. Seorang ahli desain produk dan penulis terkenal, Nir Eyal mengatakan bahwa dalam kebiasaan *hook user* yang baik melalui 4 fase *interconnected*, yaitu sebagai berikut (hal 9).

1. *Cues (or triggers)*

*Cues* merupakan informasi visual yang mendorong *user* melakukan *action* atau bertindak.

2. *Routines (or actions)*

*Routines* adalah yang dilakukan oleh para *user* pada aplikasi.

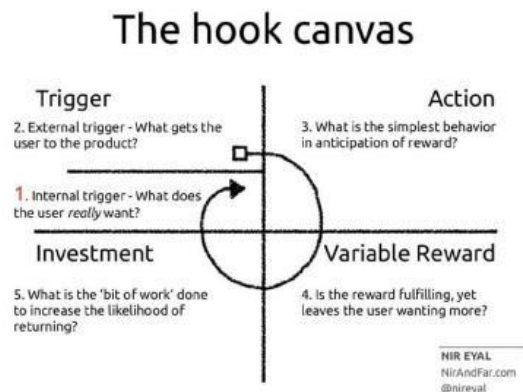
3. *Rewards*

Terdapat *Reward* nyata dan tidak nyata, *reward* nyata termasuk membuka penghargaan atau barang dan *leveling up*. Sedangkan *Reward* tidak nyata (atau berwujud) termasuk seperti bentuk visual grafik lanjutan dan *gameplay* yang menyenangkan.

4. *Investments*

Contoh pada *game*, saat *user* memperkuat karakter dan melanjutkan kelanjutan suatu *plot*, ini membuat *user* secara pribadi menginvestasi diri dalam *game event* dan hasil akhir.

Dalam merancang bentuk *gamification* perlu mengetahui apabila *action* itu berkelanjutan dan menguntungkan setelah digunakan terus-menerus, lalu didesain ke dalam format seperti pada berikut (hal 10).



Gambar 2.10. *The Hook Canvas*  
(*UX Design The Definitive Beginners Guide*, 2016)

### 2.3. *User Interface*

Pada buku *Android UI Design*, Thornsby (2016) mengatakan bahwa UI yang merupakan singkatan dari *user interface* merupakan sebuah persimpangan antara pengguna dan aplikasi atau program komputer. *User interface* adalah semua yang dapat dilihat dan dapat berinteraksi dengan *user*, semua aplikasi yang dikembangkan atau dirancang akan memiliki sebuah bentuk *user interface* (hal. 8).

#### 2.3.1. *User Interface Design*

Pada buku *Android UI Design*, Thornsby (2016) menjelaskan tahapan-tahapan dalam merancang *user interface* pada *mobile application Android* yaitu *turn ideas into sketch*, *turn sketch into wireframes*, dan *building a prototype*. Berikut merupakan penjelasan masing-masing tahapan.

##### 2.3.1.1. *Turn ideas into sketch*

*Brainstorming* merupakan langkah pertama yang perlu dilakukan adalah memikirkan beberapa *hardware* unik yang dapat ditambahkan pada *mobile*

*application*. Berikut merupakan kapabilitas *hardware Android* yang dapat dimasukkan.

A. *Touch and gestures*

*Gesture* yang dapat melakukan suatu *action* dengan cara *swiping*, *dragging*, *pinching*, dan *tapping* pada *screen* perangkat seluler.

B. *GPS*

Dapat digunakan untuk sugesti acara dan objek wisata yang berdasarkan lokasi *user*, *tag* foto-foto dengan lokasi yang sesuai, atau memberitahu lokasi *user* kepada koneksinya.

C. *Vibration*

Perlu digunakan di saat *alert audio* mengganggu, dapat digunakan juga sebagai cara lain untuk berkomunikasi dengan *user*.

D. *Audio input/output*

Penggunaan *audio input* dan *output* apabila aplikasi akan digunakan pada situasi yang mungkin akan canggung atau tidak pantas untuk berinteraksi dengan aplikasi via metode lain. Mengeluarkan *voice command* cepat pada *smartphone*.

#### E. *Interacting with other devices*

*Smartphone Android* dan *tablet* dapat terhubung melalui, contoh, *Bluetooth*, *Near Field Communication (NFC)*, *wi-fi*, dan *mobile network*. *Smartphone Android* dan *tablet* dapat terhubung dengan perangkat *Android* yang kurang tradisional dan *Google service*, seperti *wearbles*, *Android TV*, dan *beacons*.

Langkah kedua yaitu memulai proses *brainstorming*, *Write concept* untuk merancang sebuah aplikasi yang dapat menyelesaikan masalah yaitu pertama diperlukan untuk menuliskan konsep yang jelas. Sebuah konsep harus pendek dan langsung pada intinya, mulai dengan menuliskan aplikasi yang dirancang dalam satu kalimat. *Create features list*, menuliskan semua fitur yang akan ditambahkan pada aplikasi, dalam daftar fitur ini juga bisa terdapat fitur yang tidak praktis. Tambahkan semua ide fitur tanpa perlu memikirkan mengenai waktu, uang dan batasan teknis. *Identifying app's primary task*, aplikasi yang sangat efektif memiliki *primary task* yang jelas. Trik yang baik untuk masuk kedalam inti aplikasi adalah menuliskan penjelasan cepat mengenai tujuan aplikasi, suatu *statement* produk.

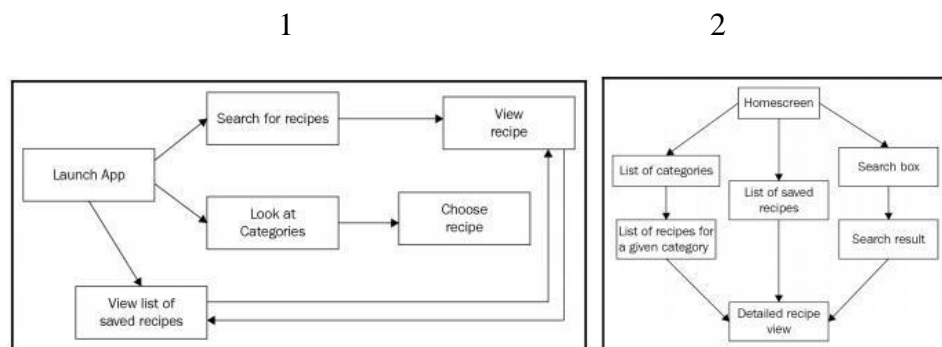
*Determining application for mobile*, setelah mendapatkan ide konten *mobile application*, perlu untuk menentukan apabila aplikasi yang dirancang cocok untuk dijalankan pada perangkat seluler atau dijalankan pada *platform* lainnya. *Budgeting*, sebelum menjadikan sebuah *mobile application*, perlu juga untuk melihat dari segi *budget* dan apakah proyek

yang dijadikan ini realistis. *Budget* termasuk uang dan juga waktu, estimasi berapa banyak waktu secara realistis dapat didedikasikan kepada proyek, dan juga gunakan informasi itu untuk menentukan berapa lama proyek itu dapat diselesaikan. Langkah ketiga yaitu *planning*, hal yang harus dipikirkan yaitu sebagai berikut. *Identifying target audience*, hal pertama yang dilakukan pada tahap *planning* adalah menentukan *target audience*, untuk “siapa” *mobile application* ditujukan atau yang butuh serta ingin memakainya. Hal yang digunakan untuk menentukan *target audience* yaitu dengan *user personas* dan *user cases*. *User personas* adalah *user* individu yang mewakili *target audience*, sedangkan *user cases* adalah skenario bagaimana, kapan, dan dimana *persona* akan memakai aplikasi.

Setelah mendapatkan rencana yang rinci mengenai seperti apa aplikasi yang akan dibuat, langkah selanjutnya yaitu *Designing*. Desain UI aplikasi agar menarik dan efektif terdapat *high-level flow*, *screen list*, *screen map*, *navigation*, *common navigation patterns*, dan *the social layer*. Langkah pertama yaitu mulai mengerjakan struktur aplikasi dengan membuat suatu *high-level flow* (gambar 1), pada tahap inilah perancang menggambarkan ide awal dari rute-rute berbeda yang *user* akan ambil pada aplikasi untuk mencapai *task* yang berbeda. *High-level flow* juga sering disebut sebagai *flowchart*. Dapat dibuat dengan *software image-editing* atau cara tradisional dengan menggambar menggunakan kertas dan pensil. Langkah selanjutnya yaitu membuat semua *screen list* yang perlu dibuat perancang apabila *user* akan bisa menyelesaikan macam-macam *task* dan

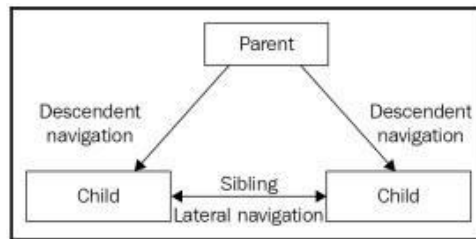


*goal*. Lalu, menggabungkan *flowchart* dan *screen list* yang telah dibuat dan membuat *screen map* yang mengekspresikan hubungan navigasi antara semua *screens* yang berbeda yang menjadikan aplikasi (gambar 2). Berikut merupakan contoh bagaimana *screen map* untuk aplikasi akan terlihat.



Gambar 2.11. *High-Level Flow* (1), *Screen Map* (2)  
(*Android UI Design*, 2016)

Aplikasi *Android* harus menjadi cukup flexibel untuk menyesuaikan dengan banyak perangkat yang berbeda, dari telepon genggam sampai *tablet* yang berukuran besar sampai konten yang ditampilkan kepada *user* dengan cara efektif. Caranya yaitu dengan *Fragments*, menggabungkan *screens* menjadi beberapa *layout* multi panel dapat meminimalisir angka interaksi yang *user* harus lakukan untuk menyelesaikan masing-masing *task*. *Navigation* pada aplikasi yang dirancang harus gampang digunakan dan intuitif, bahkan *user* baru pun dapat navigasi pada aplikasi dengan mudah. *Navigation* dibagi menjadi dua macam yaitu *descendent navigation* dan *lateral navigation*. *Descendent navigation* adalah saat *user* bernavigasi lebih dalam pada aplikasi, *user* sedang menurun dari hierarki level tinggi ke rendah. *Lateral navigation* adalah saat *user* pindah ke antar *screen*.



Gambar 2.12. *Navigation*  
(*Android UI Design*, 2016)

Saat merancang navigasi aplikasi, perancang perlu fokus pada bagaimana cara yang terbaik untuk menjadi sesuai dengan kebutuhan *target audience* dan *task* yang mereka ingin lakukan pada aplikasi. Dalam membuat struktur unik yang sesuai dengan suatu aplikasi terdapat *common navigation patterns*. Berikut adalah jenis-jenis dari *common navigating patterns*:

A. *The embedded navigation*

Umumnya dipakai pada aplikasi dengan struktur sederhana yang disematkan navigasi dalam kontennya. Navigasi ini lebih ideal untuk *screen* yang lebih besar ukurannya.

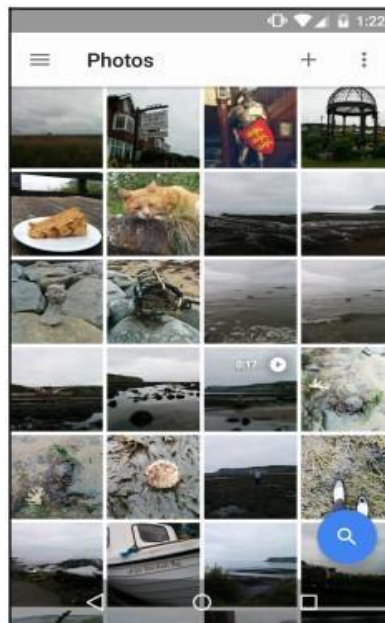
B. *Buttons and simple targets*

Merupakan navigasi yang bisa disentuh, seperti tombol pada *screen* utama aplikasi. Ini merupakan navigasi yang mudah dimengerti, akan tetapi dengan memasukkan banyak target yang dapat disentuh pada satu *screen* akan susah di navigasi pada perangkat yang lebih kecil. *Lists* digunakan saat menunjukkan banyak teks dengan cara membuat menjadi *scrolling lists*, dapat menyediakan cara navigasi aplikasi yang langsung ke inti. Akan tetapi harus berhati-hati akan daftar yang akan

terdapat anak daftar karena tipe navigasi ini dapat membuat angka *touch* yang dibutuhkan untuk menyelesaikan masing-masing *task* menjadi diluar kontrol.

### C. *Grids, cards, and carousels*

Digunakan ketika ingin menunjukkan banyak konten visual seperti *video* atau *foto*, dapat menggunakan *grid item* yang dapat di *scroll* secara vertikal atau secara horizontal untuk melihat *carousel* dan foto.



Gambar 2.13. Contoh *Grid-Based Navigation*  
(*Android UI Design*, 2016)

### D. *Tabs*

Merupakan contoh navigasi *horizontal* yang umum. Terkadang menggunakan navigasi *horizontal* lebih masuk akal, terlebih lagi apabila fitur aplikasi memiliki banyak *screen*. Navigasi ini ideal untuk

digunakan apabila pada aplikasi terdapat banyak konten yang di grup, banyak *screen*, dan set data atau konten penting yang dikategorikan.



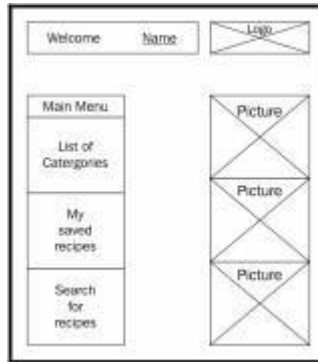
Gambar 2.14. Contoh *Tabs*  
(*Android UI Design*, 2016)

#### E. *Horizontal paging (swipe views)*

Bentuk ini sama seperti *tabs*, akan tetapi *swipe views* adalah bentuk navigasi dimana satu *screen* ditunjukkan satu per satu dan menggunakan *gesture swipe* dari kanan atau dari kiri. Bekerja dengan baik untuk aplikasi yang memiliki jumlah *screen* yang kecil.

Langkah terakhir yaitu membuat *the social layer*, perlu untuk mempertimbangkan apakah aplikasi akan memberikan manfaat dari beberapa fitur sosial, dalam ini setidaknya memperbolehkan *user* untuk dapat membagikan konten aplikasi via media sosial karena merupakan cara tercepat untuk promosi aplikasi. Terdapat juga *contacts* untuk dapat





Gambar 2.16. Contoh *Wireframing*  
(*Android UI Design*, 2016)

### 2.3.1.2. *Building a prototype*

Langkah pertama dalam pengembangan aplikasi yaitu dengan membuat *prototype digital*. *Prototyping* digital adalah saat dimana membuat rencana menjadi *test* untuk melihat apakah bekerja dengan baik di dunia nyata. Berikut merupakan langkah dalam *prototyping*.

Tahapan pertama adalah *first prototype* dengan mulai dengan membuat *prototype per screen*, biasanya dimulai dari *home screen* dan aset-aset yang terdapat pada *screen*. Referensi mengikuti *wireframe* yang telah dibuat. Tahapan kedua adalah *second prototype* dimana membuat *layout prototype* yang fleksibel dan masukkan data palsu. Data palsu ini seharusnya dapat mewakili rangka lengkap dari konten dunia nyata, *screen* ini terjamin akan ditunjukkan. Lalu tahapan terakhir adalah *finalizing design*, tahapan ini merupakan tahapan akhir dari desain aplikasi, tetapi masih belum ditambahkan *detail* dari bagaimana aplikasi akan terlihat dan membuat *user* merasakan emosi. Berikut merupakan beberapa hal yang perlu untuk dipertimbangkan ketika sedang melakukan desain akhir.

#### A. *Visuals*

Desain elemen UI aplikasi secara keseluruhan, dan harus dipastikan apabila aplikasi yang di desain merupakan desain yang menarik di mata *target audience* yang spesifik.

#### B. *Background music and sound effects*

Suara atau *sound* merupakan sesuatu yang kuat untuk mengatur suasana atau *mood*. Perlu dipertimbangkan ditambahkan atau tidak.

#### C. *Text*

Merupakan metode komunikasi yang langsung terhadap *user*. Tampilan teks memberikan dampak yang besar pada kemudahan *user* dalam membaca dan memahami apa yang ingin disampaikan. Dalam menentukan teks perlu mempertimbangkan ukuran, *style*, dan penggunaan kapital.

#### D. *Your app's personality*

Merupakan kombinasi dari banyak elemen desain yang berbeda, dari *visual*, *background music* dan nada teks.

#### E. *Creating themes and styles*

Pada tahapan inilah dimana *themes* dan *styles* menjadi berguna karena dua hal itu merupakan salah satu cara yang tercepat dan termudah dari

proses memutuskan implementasi desain pada *views*, *activities* dan bahkan seluruh aplikasi.

#### F. *Error check*

Tahap terakhir adalah memeriksa kesalahan teknis atau *error* aplikasi, *user* aplikasi akan membuat kesalahan seperti *storage space* habis, koneksi internet hilang dan kesalahan lainnya. Aplikasi yang dirancang harus dapat menangani semua kondisi dan *error* yang kurang ideal. *User input errors* adalah dimana *user* memasukkan informasi yang salah atau mengosongkan *field*, area yang paling sering terdapat *user error* seperti *login screen* atau *payment screen*.

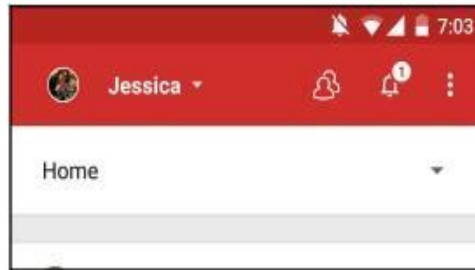
### **2.3.2. Komponen inti UI**

Thornsby (2016) menjelaskan bahwa pada *mobile application Android* terdapat komponen-komponen inti UI yang membentuk aplikasi itu sendiri, berikut terdapat komponen UI dengan penjelasan masing-masing.

#### A. *The Action Bar*

*The action bar* terdapat pada bagian atas *screen* dan biasanya berisi *action* dan elemen navigasi seperti pada gambar berikut.

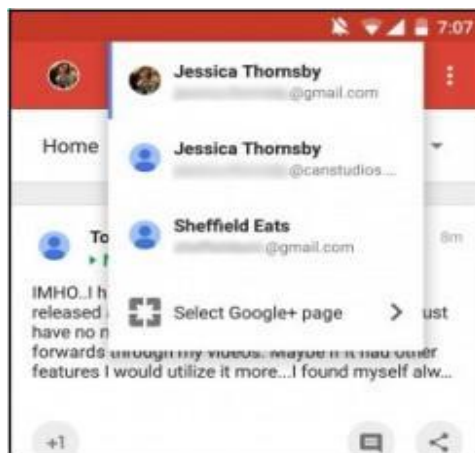




Gambar 2.17. Contoh *The Action Bar*  
(*Android UI Design*, 2016)

## B. *Navigational Controls*

*Navigational controls* terdapat yang mudah seperti *the standard Back button*, atau dapat menjadi lebih rumit seperti *view-switching controls*.



Gambar 2.18. Contoh *Navigational Controls*  
(*Android UI Design*, 2016)

## C. *Action Buttons*

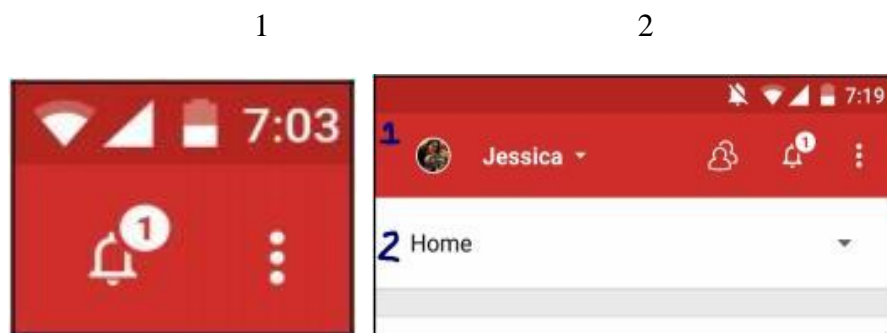
*Action bar* ideal digunakan untuk menyediakan akses dalam *action* terpenting pada aplikasi dengan mudah berdasarkan konteks sementara *user*. *Action* yang muncul pada *action bar* dikenal sebagai *action buttons*. Pada gambar berikut merupakan salah satu contoh dari *action buttons*.



Gambar 2.19. Contoh *Action Buttons*  
(*Android UI Design*, 2016)

#### D. *The Action Overflow*

*Action overflow* digunakan di saat *action buttons* tidak muat pada *action bar*, sistem *Android* akan secara otomatis memindahkan beberapa dari *action buttons* kepada *action overflow* (gambar 1). Pada *action overflow*, terdapat juga *split action bar* yang digunakan di saat ingin menampilkan jumlah *action* yang banyak tanpa membuat semua *action* itu berakhir dimasukkan pada *action overflow* pada *screen* yang lebih kecil (gambar 2).

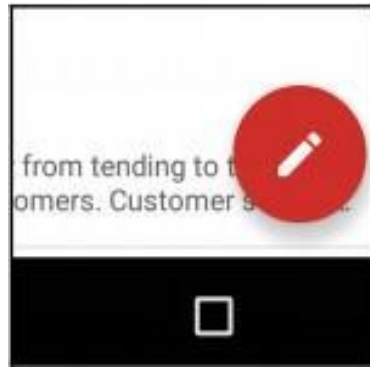


Gambar 2.20. Contoh *Action Overflow* (1), *Split Action Bar* (2)  
(*Android UI Design*, 2016)

#### E. *Floating Action Button*

*Floating action button* atau disingkat sebagai FAB merupakan tombol sirkular yang melayang diatas *user interface* utama. FAB ini memudahkan aplikasi

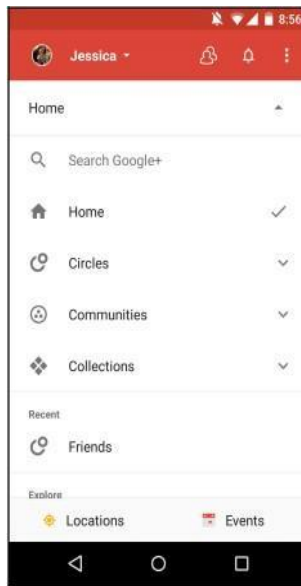
dalam menyoroti *action* primer pada konteks, pada gambar berikut merupakan FAB.



Gambar 2.21. *Floating Action Bar*  
(*Android UI Design*, 2016)

#### F. *Menus*

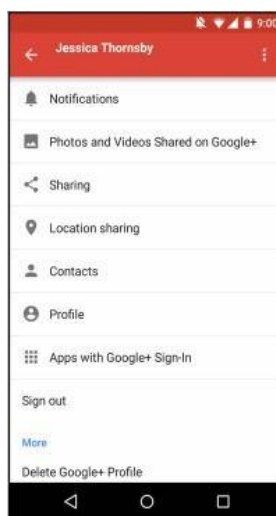
Menu merupakan salah satu cara utama dalam bernavigasi pada aplikasi dan juga merupakan komponen yang harus ada dalam sebagian banyak UI *Android*. Menu menyediakan *user* dengan daftar opsi yang dapat dipilih dan umumnya berupa satu kata atau satu baris teks seperti pada gambar berikut.



Gambar 2.22. Menu  
(*Android UI Design*, 2016)

## G. *Settings*

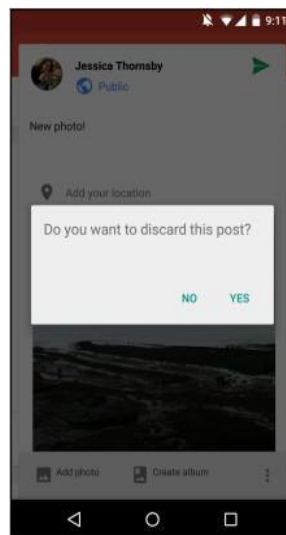
*Settings* merupakan komponen tetap pada banyak aplikasi *Android* karena berfungsi untuk mengatur perilaku aplikasi untuk menyesuaikan dengan para *user*.



Gambar 2.23. Contoh *Settings*  
(*Android UI Design*, 2016)

## H. *Dialogues*

*Dialogues* merupakan elemen UI yang dapat membantu aplikasi dalam menarik perhatian *user*, *Dialogue* adalah *window* kecil yang dapat berisi teks dan tombol. Tujuan *dialogues* adalah memberikan informasi penting seperti *terms* dan *conditions* atau suatu *disclaimer*, menanyakan permohonan informasi tambahan seperti, *dialogue* yang meminta *user* untuk memasuki password, meminta *user* untuk menentukan pilihan seperti menghapus atau menyimpan *draft* status.



Gambar 2.24. Contoh *Dialogue*  
(*Android UI Design*, 2016)

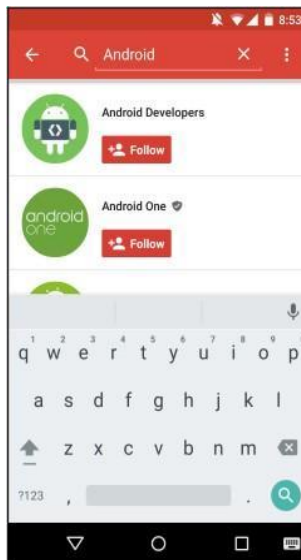
## I. *Toasts*

*Toast* adalah *popup* kecil yang menyediakan *user feedback* sederhana. Berbeda dengan *dialogue*, *toast* tetap aktif selama ia masih terlihat pada aplikasi. Memilih *toast* dibandingkan *dialogue* ketika ingin mengirimkan pesan pendek

dan sederhana kepada *user* dan terlebih lagi apabila tidak meminta *user* untuk *input* sesuatu.

## J. Search

Sebagian besar UI terdapat suatu bentuk *search* untuk *user* dapat mencari data dengan mudah dan cepat. Berikut merupakan contoh komponen UI *search*.



Gambar 2.25. Search  
(Android UI Design, 2016)

Pada *Android* terdapat dua macam *framework search* standar yang dapat diimplementasikan pada aplikasi. *Search dialogue* biasanya disembunyikan, ketika diaktivasi oleh *user* akan muncul sebagai *search box* melayang pada atas *activity* saat itu. Komponennya dikontrol oleh sistem *Android*. *Search widget* merupakan suatu contoh dari *SearchView* yang dapat diposisikan pada area manapun pada UI aplikasi.

#### A. *Input controls*

*Input controls* merupakan variasi komponen yang disediakan oleh *platform Android* yang dapat berinteraksi dengan *user*.

#### B. *Styles and themes*

*Styles* dan *themes* merupakan hal yang mengikat semua elemen dan memastikan UI terasa konsisten. Berikut merupakan perbedaan antara *style* dan *theme*. *Style* merupakan koleksi dari properti yang dapat dimasukkan pada seluruh aktivitas atau aplikasi, atau pada *view* individual. *Theme* merupakan *style* yang dapat dimasukkan pada seluruh aktivitas atau aplikasi, tetapi bukan pada *view* individual.

### **2.4. Elemen desain**

Menurut Landa (2016) dalam buku *Graphic Design Solution* edisi ke-5, desain grafis merupakan suatu bentuk dari visual komunikasi yang digunakan sebagai penyampaian suatu pesan atau informasi kepada audience. merupakan suatu ide yang berupa visual yang mengandalkan pada kreasi, seleksi, dan organisasi dari elemen-elemen visual.

#### **2.4.1. *Formal elements***

Menurut Landa (2016) *formal elements* dari bentuk 2 dimensi yaitu garis, bidang, warna, dan tekstur. Penjelasan masing-masing elemen terdapat pada berikut.

#### **2.4.1.1. Garis**

Garis merupakan unit terkecil yang umumnya dikenal sebagai sirkular. Pada gambar yang berbasis pada *screen*, suatu titik adalah sesuatu yang terlihat, sebuah *pixel* cahaya (dengan ataupun tanpa *hue*) yang berbentuk kotak. Garis dapat memandu yang melihat pada suatu arah. Suatu garis dapat memiliki suatu kualitas spesifik, dapat menjadi halus atau keras, mulus atau terputus, tebal atau tipis, teratur atau berubah (hal. 19). Berikut merupakan macam-macam kategori dari garis.

- A. *Solid line* : merupakan sebuah tanda yang digambar pada atas suatu permukaan.
- B. *Implied line* : merupakan garis tidak berlanjut yang dilihat sebagai berlanjut.
- C. *Edges* : suatu pertemuan titik atau garis pembatas antara bidang dan *tones*.
- D. *Line of vision* : suatu gerakan mata seseorang saat sedang mengamati suatu komposisi atau juga disebut sebagai garis dari gerakan atau suatu garis yang terarah.

#### **2.4.1.2. Bidang**

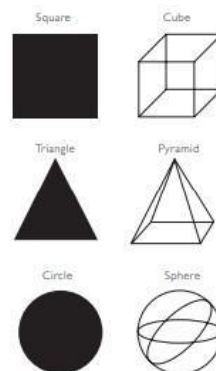
Bidang merupakan garis luar umum pada sesuatu dan terbentuk sebagai suatu bentuk atau jalan tertutup. Merupakan area yang digambarkan pada permukaan 2 dimensi dengan garis (garis luar, kontur) atau dengan warna, *tone*, atau tekstur. Berikut merupakan salah satu contoh dari bidang yang terbentuk dengan warna (hal. 20).





Gambar 2.26. *Shapes by Color*  
(*Android UI Design*, 2016)

Bidang pada dasarnya datar, dua dimensi dan dapat diukur dengan tinggi dan lebar. Pada dasarnya semua bidang dapat terbentuk dari tiga penggambaran dasar yaitu: kotak, segitiga, dan bulat. Masing-masing dari bidang dasar memiliki bentuk dengan *volume* atau padat yang sesuai: *cube*, *pyramid*, dan *sphere* (hal. 21).



Gambar 2.27. *Basic Shapes and Forms*  
(*Android UI Design*, 2016)

Terdapat juga yang dinamakan *figure* atau *ground*, ini juga disebut dengan ruang positif dan negatif. Ini merupakan prinsip dasar dari persepsi visual dan mengacu pada hubungan antar bidang, pada *figure* terhadap *group*, pada permukaan dua dimensi. Pada bidang juga terdapat bentuk

*letterforms, numerals, dan punctuation marks. Type* juga dapat berbentuk *rectilinear, curvilinear, geometric*, atau *organic* dan bentuk tipografi bisa di generate pada komputer atau juga dibuat dengan tangan. Bentuk *letterform, numeral, atau punctuation mark* termasuk *figure*, sedangkan *counters* atau ruang terbuka *type* termasuk *ground* atau *negative space* (hal. 22).

#### **2.4.1.3. Warna**

Warna merupakan suatu properti atau deskripsi dari energi cahaya, hanya dengan cahaya membuat orang dapat melihat warna. Warna pada suatu permukaan objek dilihat dan diketahui sebagai cahaya atau warna yang memantul. Warna terdapat pigmen yang merupakan zat kimia pada suatu objek yang berinteraksi dengan cahaya untuk mengetahui karakteristik warna yang didapat, sesuatu yang memiliki warna dasar atau pigmen yang dibuat secara artifisial ditambahkan pada agen warna seperti kertas, tinta, dan plastik. Permukaan yang secara natural memiliki pigmen dilihat sebagai cahaya yang memantul, sedangkan warna pada layar komputer adalah suatu energi cahaya (*Wavelength*) yang bisa disebut dengan warna digital. Contoh ketika memilih warna biru murni pada *Adobe Photoshop™*, warna yang akan terlihat adalah *wavelength* cahaya biru itu sendiri. Warna digital yang dilihat pada layar disebut sebagai warna *additive*.

Dalam warna terdapat tiga kategori elemen warna, yaitu: *hue, value,* dan *saturation. Hue* adalah nama dari warna, seperti merah, hijau, biru atau oranye. *Value* merupakan level *luminosity*, terang atau gelapnya suatu

warna. *Saturation* adalah kecerahan atau kekusaman pada suatu temperatur warna, mengacu pada apabila warna terlihat hangat atau dingin. Dalam warna terdapat warna primer yang merupakan warna dasar. Pada saat bekerja dengan cahaya pada media berlayar, terdapat tiga warna primer yaitu *red, green, and blue* (RGB). Warna primer ini juga disebut sebagai primer *additive* karena ketika digabungkan semua menjadi jumlah yang sama, warna *red, green* dan *blue* menciptakan cahaya putih (hal. 23).



Gambar 2.28. *Additive Color*  
(*Android UI Design*, 2016)

#### 2.4.1.4. **Tekstur**

Tekstur merupakan kualitas taktil pada suatu permukaan atau simulasi atau perwakilan suatu kualitas permukaan, dalam tekstur terbagi menjadi dua kategori dari tekstur: taktil dan visual. Tekstur taktil memiliki kualitas taktil dan dapat disentuh dan dirasakan secara fisik, ini juga disebut sebagai tekstur sebenarnya. Tekstur *visual* merupakan ilusi dari tekstur asli yang dibuat dengan tangan, di *scan* dari tekstur asli, atau difoto.

*Pattern* merupakan pengulangan konsisten dari suatu unit visual atau elemen pada area yang tersedia. Pada *patterns* terdapat struktur yang

bergantung pada konfigurasi dari tiga blok bangunan, yaitu: *dots*, *lines*, dan *grids* (Hal. 28).

#### 2.4.2. Semiotika

Menurut Landa (2016) semiotika merupakan studi mengenai *signs* dan simbol, Berikut merupakan klasifikasi dari *sign*.

1. *Sign* : suatu tanda visual atau bagian dari bahasa yang menunjukkan hal lain, seperti kata binatang anjing dan *pictograph* anjing yang keduanya mewakili “anjing”.
2. *Icon* : merupakan suatu visual, gambar atau simbol *pictorial* yang mewakili obyek, *action*, dan konsep. *Icon* menyerupai sesuatu yang diserupai atau membagi suatu kualitas dengannya, sesuatu itu bisa suatu fotografi, perwakilan *pictorial*, elemen visual, *arbitrary*, atau simbolis.
3. *Index* : suatu *sign* yang menandakan sesuatu melalui hubungan langsung antara *sign* dan obyek tanpa mendeskripsikan atau menyerupai sesuatu yang diserupai. *Index* dapat terjadi dalam cara yang bervariasi: dapat sebagai *cue* yang membuat yang melihat memikirkan referensi, dengan kedekatannya, dengan menunjukkan kepada sesuatu yang ditandakan, atau dengan menjadi bukti fisik.
4. *Symbol* : suatu visual yang memiliki *arbitrary* atau hubungan yang konvensional antara yang menandakan dan sesuatu yang ditandakan. Dalam membuat simbol, memecahkan arti dari sesuatu melalui asosiasi yang

dipelajari. Sebagai contoh, burung merpati memiliki arti “diterima” sebagai simbol perdamaian. Selain itu, kata-kata yang ditulis maupun dibaca juga merupakan simbol (hal. 116).



Gambar 2.29. Simbol *Peace*  
(*Android UI Design*, 2016)

### **2.4.3. *Typography***

Menurut Landa (2016) *typography* adalah seni dan teknik dalam mengatur *type* untuk membuat suatu kalimat yang ditulis yang dapat dibaca, dipahami, dan juga menarik saat ditampilkan. Dalam *typography* terdapat elemen dari *type* yaitu disebut sebagai *typeface*, merupakan desain dari satu set karakter yang digabungkan dengan properti visual yang konsisten. Properti visual dengan *style* menciptakan karakter yang harus ada pada suatu *typeface* yang tetap dapat dikenal meskipun *typeface* telah dimodifikasi. Pada umumnya, suatu *typeface* termasuk *letters*, *numbers*, *symbols*, *signs*, *punctuation marks*, dan *accent* atau tanda diakritik (hal. 44). Dalam *type* memiliki beberapa klasifikasi yaitu sebagai berikut.

#### **2.4.3.1. *Classification of types***

A. *Old style* : *typeface* roman, merupakan keturunan langsung dalam bentuk dari huruf yang digambar dengan pena tinta. Diberi

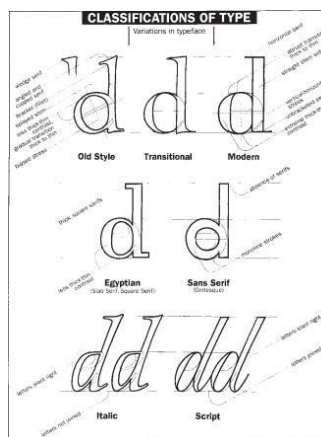
karakter dengan *serif* yang miring dan mengurung. Sebagai contoh adalah *Caslon*, *Garamond*, *Hoefler Text*, dan *Times New Roman*.

- B. *Transitional* : merupakan *typeface serif*, mewakili transisi dari *style* lama menjadi *modern* dan memiliki karakteristik kedua desain. Contoh seperti *Baskerville*, *Century*, dan *ITC Zapf International*.
- C. *Modern* : merupakan *typeface serif*, memiliki bentuk yang lebih geometris dalam segi konstruksi yang berlawanan dengan *typeface Old Style*. memiliki karakteristik kontras coretan tebal-tipis dan *vertical stress*. Beberapa contoh dari *typeface Modern*, yaitu *Didot*, *Bodoni*, dan *Walbaum*.
- D. *Slab Serif* : *typefaces serif*, memiliki karakteristik *serif* yang berat dan seperti bentuk lempeng. merupakan sub-kategori *Egyptian* dan *Clarendon*. Ini yang merupakan beberapa contoh dari *typeface Slab serif*: *American Typewriter*, *Memphis*, *ITC Lubalin Graph*, *Bookman*, dan *Clarendon*.
- E. *Sans Serif* : *typeface* ini memiliki karakteristik seperti *typeface serif*, contoh *typeface* ini merupakan *Futura*, *Helvetica*, and *Univers*. Terdapat juga bentuk huruf tanpa *serif* yang memiliki coretan tipis dan tebal, contoh dari ini merupakan *Grotesque*, *Franklin Gothic*, *Universal*, *Futura*, and *Frutiger*. *Sans serif typeface subcategories include Grotesque, Humanist, Geometric*, dan lainnya.
- F. *Blackletter* : *typeface* ini juga disebut dengan *gothic*, karena memiliki karakteristik coretan yang berat dan menambahkan beberapa

lekukan. Contoh dari *typeface* ini adalah *Rotunda*, *Schwabacher*, dan *Fraktur*.

G. *Script* : huruf *typeface* ini pada umumnya berupa seperti lempeng dan tergabung, *type Script* dapat meniru bentuk hasil tulisan pena *chisel-edged*, pena fleksibel, pena *pointed*, pensil, atau kuas. Contoh dari *typeface* ini adalah *Brush Script*, *Shelley Allegro Script*, dan *Snell Roundhand Script*.

H. *Display* : *typeface* ini didesain untuk keperluan huruf dengan ukuran besar, lebih sering *typeface* ini digunakan untuk *headline* dan judul. Bentuk *typeface* ini lebih menjelaskan, terhias, buatan tangan, dan dapat masuk ke dalam klasifikasi lainnya (hal. 47).



Gambar 2.30. *Classification of Type Chart*  
(*Android UI Design*, 2016)

#### 2.4.4. *Layout*

Menurut Landa (2016) menjelaskan bahwa, *grid* adalah sebuah panduan struktur berkomposisi yang terbuat dari vertikal dan horizontal yang membagi suatu *format*

menjadi kolom dan *margin*. Struktur dari buku, majalah, brosur, *desktop website*, *mobile website*, dan lainnya mempertegaskan *grid*.

Menurut Thornsby (2016) mengatakan bahwa *density-independent pixel* adalah unit abstrak yang berbasis pada *density* fisik layar. Unit dp relatif pada satu *pixel* fisik pada layar 160 *dots per inch*. Saat dijalankan, *Android* akan secara otomatis menyesuaikan angka atau jumlah *pixel* untuk menggambar 1 dp dengan faktor yang sesuai untuk dp *screen* yang sedang dipakai. Dengan menggunakan ukuran *density-independent* merupakan solusi yang cepat dalam menciptakan suatu UI yang dapat menyesuaikan dengan berbagai ukuran *screen* secara otomatis.

## **2.5. Nutrisi Anak Balita**

Pada buku Gizi dalam Daur Ulang, Damayanti, Pritasari, dan Nugraheni (2017) mengatakan bahwa dalam daur atau siklus kehidupan manusia, pada saat anak yang masih berusia bayi sampai remaja merupakan masa pertumbuhan yang tercepat. Berat badan bayi yang sehat akan naik sebesar 300% pada usia 1 tahun, sedangkan pada usia remaja berat badan dapat naik sebesar 50% dan tinggi badan dapat naik sebesar 20%. Agar anak memiliki masa pertumbuhan yang baik, mereka memerlukan makanan yang bernutrisi atau dengan gizi yang cukup (hal. 2).

Dijelaskan bahwa kondisi kesehatan pada satu tahap dapat dipengaruhi oleh tahap sebelumnya. Sebagai contoh yaitu, seorang anak balita perempuan dengan Kurang Energi Protein (KEP) akan berpotensi tumbuh menjadi anak remaja dengan gangguan pertumbuhan atau Kurang Energi Kronik (KEK) yang berisiko menjadi ibu hamil yang KEK (hal. 3). Terdapat dua faktor utama yang mempengaruhi



keadaan kesehatan secara langsung pada setiap individu, yaitu konsumsi makanan dan adanya penyakit infeksi. Sedangkan pengaruh tidak langsung merupakan ketersediaan makanan di keluarga, asuhan ibu dan anak serta pelayanan kesehatan. Maka dari itu, saat anak masih usia balita sangat diperlukan untuk mengonsumsi makanan dengan nutrisi yang cukup karena dapat mempengaruhi pertumbuhannya sampai usia dewasa atau tua (hal. 4).

Dalam memenuhi kebutuhan nutrisi yang cukup yang sesuai dengan masing-masing individu, menggunakan Angka Kecukupan Gizi (AKG), Tabel Komposisi Pangan Indonesia (TKPI) dan juga dengan penggunaan Daftar Bahan Makanan Penukar (DBMP). Angka Kecukupan Gizi (AKG) merupakan suatu kecukupan rata-rata zat gizi setiap hari individu yang berdasarkan golongan umur, jenis kelamin, ukuran tubuh, aktivitas tubuh untuk mencapai derajat kesehatan yang optimal (hal. 8). TKPI merupakan *database* yang menggabungkan semua data komposisi zat gizi makanan Indonesia menjadi satu buku, data komposisi bahan makanan ini memiliki berbagai jenis nama yaitu Daftar Komposisi Bahan makanan (DKBM) (hal. 16). DBMP merupakan suatu daftar yang berisi daftar nama bahan makanan, berat dalam ukuran rumah tangga (URT), berat dalam gram serta kandungan energi, protein, karbohidrat dan lemak dari makanan tersebut (hal. 23).

Anak balita merupakan anak yang berusia 1-3 tahun dan 4-5 tahun yang termasuk anak prasekolah (hal. 211). Berikut merupakan AKG yang dianjurkan bagi anak yang berusia 1 sampai 3 tahun dan berusia 3 sampai 5 tahun di Indonesia, terdapat empat kolom dan seterusnya berisi kecukupan energi dan zat gizi sehari

untuk kelompok umur. Zat gizi yang dicantumkan terdiri dari zat gizi makro yaitu karbohidrat, protein, lemak, serat dan air, serta vitamin dan mineral.

Tabel 2.2. Angka Kecukupan Energi, Protein, Lemak, Karbohidrat, Serat dan Air Yang Dianjurkan untuk Per Hari Anak Balita Indonesia

No.	Nutrisi	1-3 tahun	4-5 tahun
1	Energi	1125 kkal	1600 kkal
2	Protein	26 gr	35 gr
3	Lemak total	44 gr	62 gr
4	Lemak n-6	7,0 gr	10,0 gr
5	Lemak n-3	0,7 gr	0,9 gr
6	Karbohidrat	155 gr	220 gr
7	Serat	16 gr	22 gr
8	Air	1200 ml	1500 ml

Tabel 2.3. Angka Kecukupan Vitamin Larut Lemak yang dianjurkan untuk per hari anak balita Indonesia

No.	Nutrisi	1-3 tahun	4-5 tahun
1	Vitamin A	400 mcg	450 mcg
2	Vitamin D	6 mg	7 mg
3	Vitamin E	6 mg	7 mg
4	Vitamin K	15 mcg	20 mcg

Tabel 2.4. Angka Kecukupan Mineral Makro Yang Dianjurkan untuk Per Hari Anak Balita Indonesia

No.	Nutrisi	1-3 tahun	4-5 tahun
1	Kalsium	650 mcg	1000 mcg
2	Fosfor	500 mcg	500 mcg
3	Magnesium	60 mg	95 mg
4	Natrium	1000 mg	1200 mg
5	Kalium	3000 mg	3800 mg
6	Mangan	1,2 mg	1,5 mg

Tabel 2.5. Angka Kecukupan Mineral Mikro yang dianjurkan untuk per hari anak balita Indonesia

No.	Nutrisi	1-3 tahun	4-5 tahun
1	Tembaga	340 mg	440 mg
2	Kromium	11 mcg	15 mcg
3	Besi	8 mg	9 mg
4	Iodium	120 mcg	120 mcg
5	Seng	4 mg	5 mg
6	Selenium	17 mcg	20 mcg
7	Fluor	0.6 mg	0.9 mg