

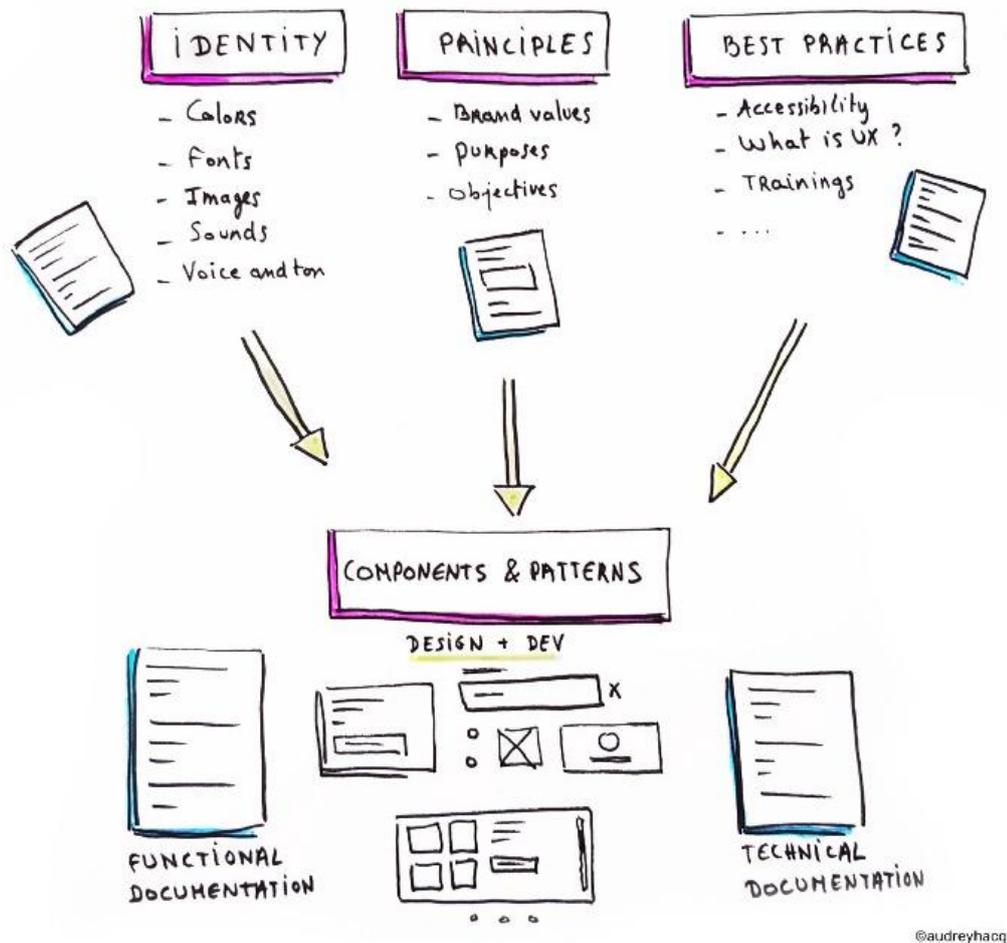
## BAB 2

### LANDASAN TEORI

#### 2.1 Design System

*Design System* adalah sebuah kumpulan komponen-komponen UI dan *code* yang saling terhubung dan mempunyai kapabilitas *reusable* yang tinggi, dipandu oleh standar yang jelas dan terintegrasi dalam suatu *system* untuk mempercepat dan mempermudah proses perancangan desain dan *development*. Tujuan *design system* adalah untuk membuat standarisasi yang terdokumentasi, meliputi *pattern library*, *code library* dan *styleguide*, sehingga dapat digunakan oleh seluruh tim *development* [2].

Domumentasi *Design System* diantaranya terdiri dari *brand identity*, *design principle*, *pattern library*, *design/shared language* dan *code library* [4].



Gambar 2-1 Design System Document



*Voice and tone* harus bisa menuntun bagaimana nada pada suatu perusahaan dapat beradaptasi dengan berbagai situasi. Seperti yang MailChimp contohkan, misalnya pengguna tidak bisa menggunakan kartu kredit karena ditolak, maka *voice* and *tone* dari produk tersebut harus memberikan kesan nada yang serius dan lebih jelas mengapa kartu kredit pengguna tersebut tidak dapat digunakan [4].

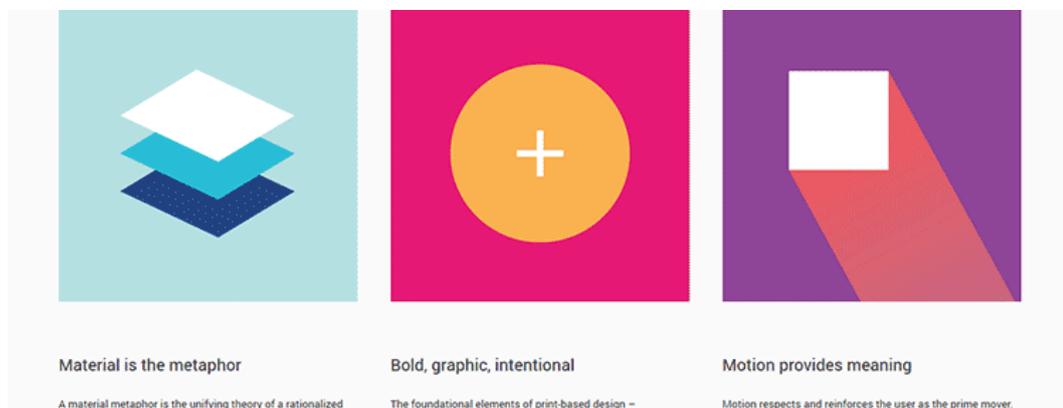


**Gambar 2-3 Proses Penentuan Voice and Tone**

### 3.7.3 Design Principle

*Design principle* adalah sekumpulan nilai yang berfungsi sebagai kompas suatu produk. *Design principle* menjadi sebuah aturan yang disepakati oleh seluruh tim agar bisa dijalur yang sama dan bergerak melalui proses desain yang sama.

*Design principle* harus spesifik, bernuansa dan dapat ditindaklanjuti [10]. *Google material design* menjadi produk yang kohesif, dikenal banyak orang. Dalam dokumentasi-nya *material design* mendefinisikan filosofi desain, tujuan, prinsip-prinsip umum dan menyediakan aplikasi spesifik dari *material design* [2].



**Gambar 2-4 Google Material Design Principle**

### 3.7.4 Desain/Shared Language

Bisakah memastikan suatu produk masih terasa kohesif dan utuh, bahkan ketika banyak orang yang mengerjakannya? Jawabannya Ya, jika seseorang memiliki pemahaman bersama tentang apa itu “*Design System*” dan bagaimana cara kerjanya. Ini berarti bahwa tim produk mengikuti prinsip panduan yang sama, bahwa visi merek sejalan, memiliki pendekatan bersama untuk desain dan arsitektur *front-end*, dan mengetahui pola yang paling efektif dan apa yang membuatnya berhasil. Dengan kata lain, membuat *design system* yang kohesif membutuhkan bahasa bersama [4].

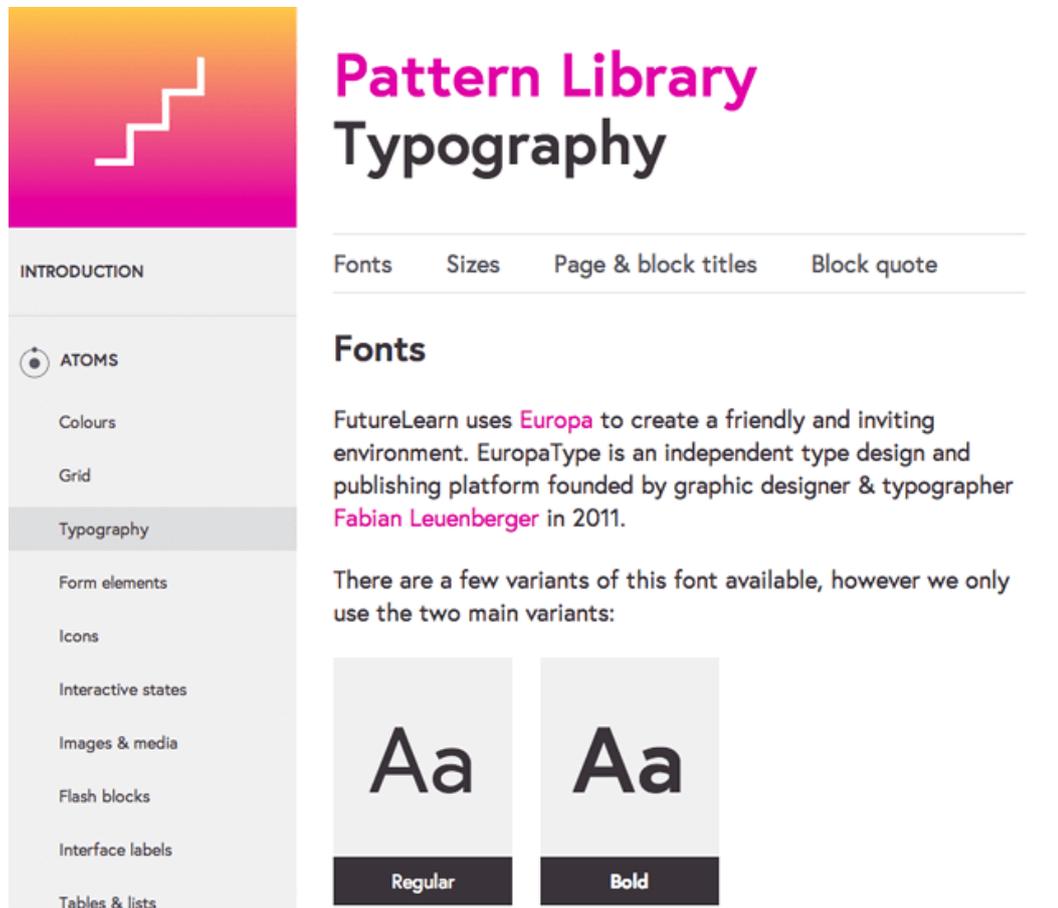
Bahasa sangatlah penting untuk kolaborasi. tanpa adanya *shared language* sekelompok orang tidak dapat berkreasi secara efektif. karna setiap orang memiliki pemikiran dan mental model yang berbeda dari apa yang ingin mereka capai. sehingga orang-orang yang terlibat dalam pembuatan produk dalam sebuah tim perlu dibagikan *share language* [4].

Dalam buku *How to Make Sense of Any Mess*, AbbyCovert menyarankan bahwa *share language* harus dibuat sebelum memikirkan tentang *user interface* yang akan dibuat, dengan cara berdiskusi, memeriksa dan mendokumentasikan keputusan bersama. Hal ini diterapkan untuk menggambarkan konsep serta bahasa sehari-hari yang digunakan untuk berbicara tentang keputusan desain. Memiliki bahasa bersama berarti memiliki pendekatan yang sama untuk menamai elemen, mendefinisikan pola desain, ataupun nama yang digunakan dalam desain [12].

### 3.7.5 Pattern Library

*Pattern library* adalah *tool* untuk menyimpan, mengumpulkan dan membagikan pola desain. *Pattern library* menjadi prinsip atau pedoman untuk membuat standar dokumentasi pada *design system*. *Pattern library* dipengaruhi oleh *style guide* dan juga dikenal dengan *component library* [4]. *Components* memiliki kompleksitas yang beragam. Mengurangi *Components* menjadi satu fungsi, seperti tombol atau *drop-down* bisa meningkatkan fleksibilitas, yang membuatnya lebih dapat digunakan kembali. Semakin banyak komponen dapat digunakan kembali, maka semakin sedikit yang perlu di *maintain*, dan semakin mudah skalanya [9].

*Pattern library* awalnya hanya untuk identitas produk, seiringnya waktu *pattern library* dimanfaatkan untuk pembentukan *typography*, *color palettes*, dan komponen antarmuka lainnya [9].



**Gambar 2-5 FutureLearn Pattern Library**

### **3.7.6 Code Library**

*Code Library* merupakan sekumpulan *code* yang memiliki fungsi-fungsi tertentu dan mempunyai *reusable* atau *code* yang dapat dipanggil kembali kedalam program lain. Adanya *code library* dibuat untuk mempermudah dalam membangun sebuah *software* dan *programers* tidak harus membuat *code* dari awal untuk suatu fungsi tertentu [11].



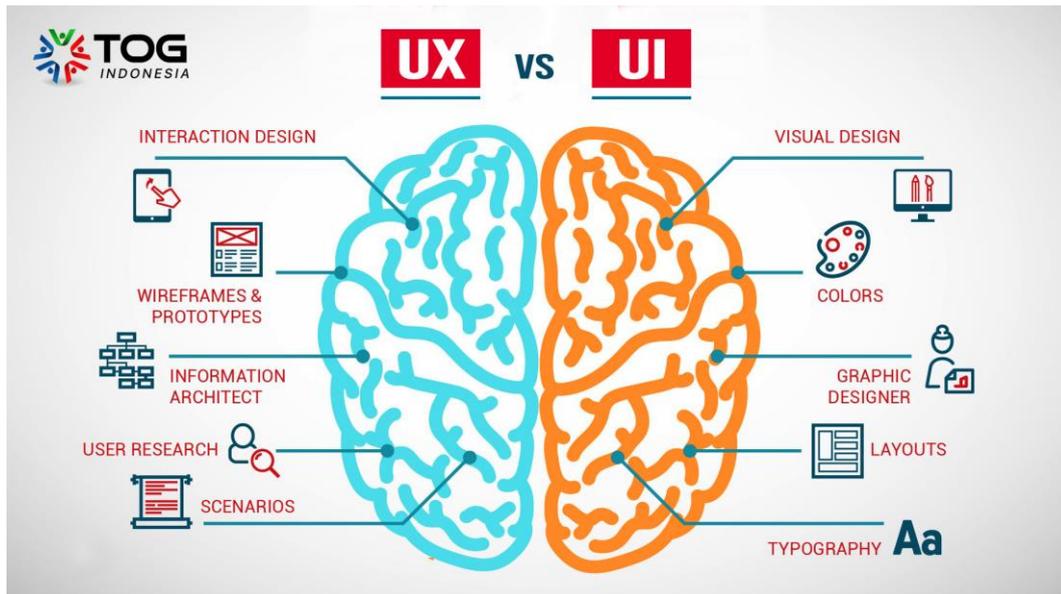
**Gambar 2-6 Code Library**

### 3.7.7 Style Guide

*Style guide* adalah kumpulan modular semua elemen pada *User Interface* (UI) yang mempunyai *reusable*. Mulai dari aturan *branding* hingga jumlah untuk tombol *call to action*. *Style guide* juga sering merujuk pada aturan dan praktik terkait hal yang boleh dan tidak boleh dilakukan pada desain yang dibangun. Pada tingkat yang lebih holistik, *Style guide* juga merupakan tempat yang tepat untuk menyatakan motif dan pernyataan misi sebuah perusahaan [13].

## 2.2 User Experience dan User Interface

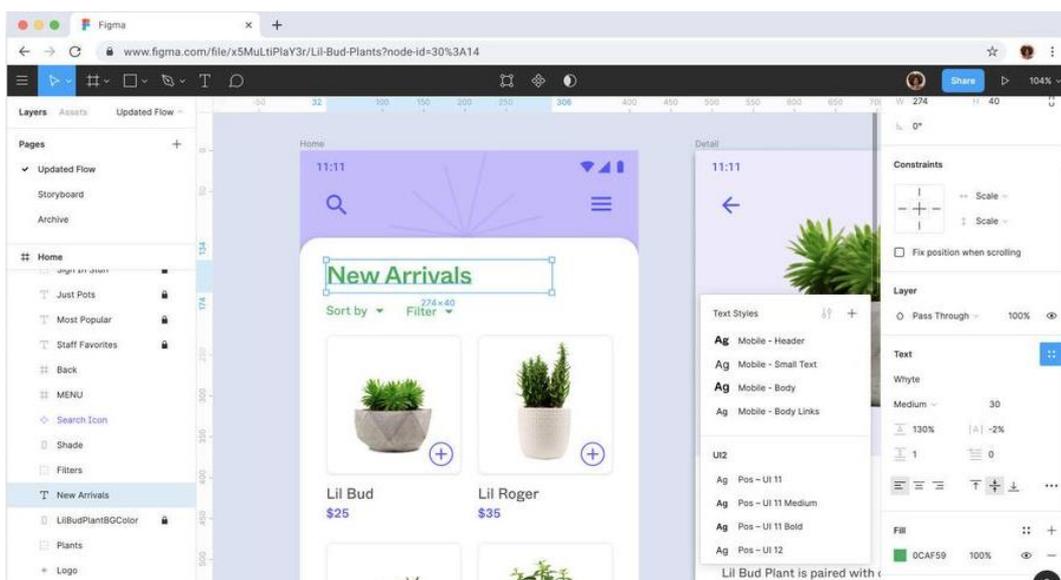
*User experience* yang baik akan menciptakan pengalaman menyenangkan bagi penggunanya. hadirnya *user experience* membuat pengguna menjadi lebih nyaman dan mudah saat menggunakan produk. Dapat dikatakan UX adalah apa yang dirasakan oleh pengguna seperti ketertarikan, kemudahan, kenyamanan dan lain lain. Sedangkan apa yang dilihat, didengar dan disentuh oleh pengguna itu adalah *user interface* [8]. Adapun perbedaan UX dan UI bisa dilihat pada gambar.



Gambar 2-7 Perbedaan UX dan UI

### 2.3 Figma Design Tool

Figma adalah sebuah *flatfrom protoyting online* dan kolaboratif, dimana desain antarmuka dapat dikerjakan oleh lebih dari satu orang secara bersama-sama secara *real-time* walaupun ditempat yang berbeda [19]. kemampuan aplikasi figma tersebutlah yang membuat *tool* ini menjadi pilihan banyak UI/UX desainer untuk membuat *prototype* website maupun aplikasi dengan waktu yang cepat dan efektif [20].



Gambar 2-8 Tampilan design tool figma

## 2.4 Standard Design

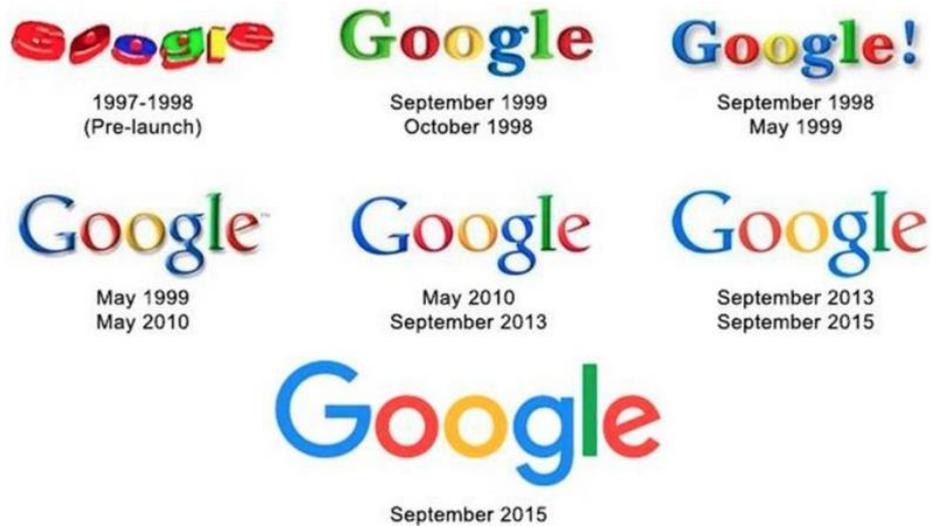
*Standard* menjadi bagian penting pada suatu *design system*. Dengan adanya *standard* di *design system* mampu menciptakan pengalaman pengguna yang jauh lebih baik. Mendefinisikan dan mematuhi standar adalah cara bagaimana menciptakan pemahaman itu. Melakukannya akan menghilangkan subjektivitas dan ambiguitas yang sering kali menimbulkan perdebatan, gesekan dan kebingungan dalam tim *development* [9].

Pembentukan *components* harus diikuti dengan *standard* yang dibuat agar *components* bisa digunakan secara konsisten dan berulang, membuat aplikasi yang lebih dapat diprediksi dan mudah dipahami. Selain itu juga memungkinkan desainer untuk menghabiskan lebih sedikit waktu berfokus pada *style* dan lebih banyak waktu untuk mengembangkan *experience* pengguna yang lebih baik [9].

## 2.5 Consistency

Pembuatan *style guide* dan *components* yang konsisten dan kohesif akan memberikan keuntungan yang baik bagi orang yang membuat dan penggunanya [9]. Pada tahapan proses analisis sampel website di Apookat software house, banyak ditemukan *style* dan *components* yang tidak konsisten dan kohesif seperti halnya *card*, dimana *style card* yang digunakan dalam satu halaman ada yang menggunakan *hover* dan ada yang tanpa *hover*, padahal keduanya sama sama tidak ada *action click*. Hal itu membuat pengguna bingung dan bertanya.

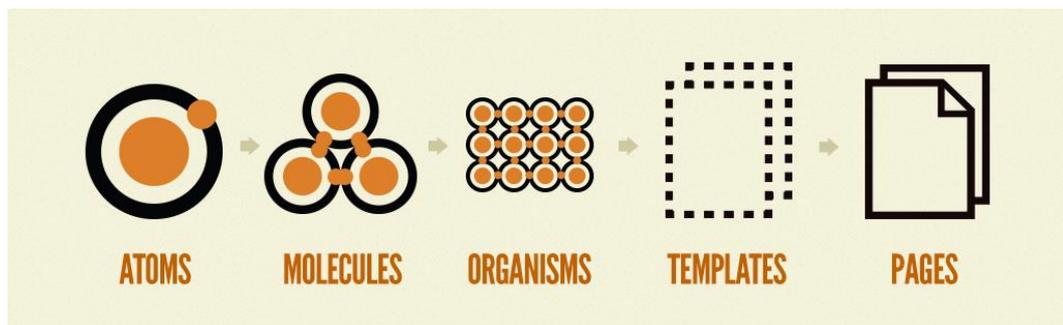
*Consistency* bertujuan untuk meningkatkan kemampuan belajar ketika pengguna menemukan elemen desain yang serupa dan memiliki fungsi yang sama. Selain itu *consistency* sangat penting untuk mengurangi *load cognitive* pada tampilan antarmuka [14].



**Gambar 2-9 Consistency Google Logos**

## 2.6 Atomic Design Methodology

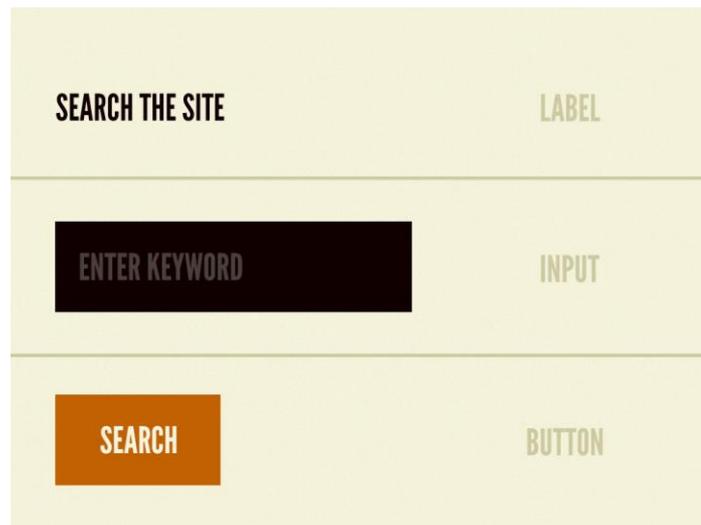
*Atomic design methodology* merupakan suatu model yang terbentuk dari kumpulan *components* yang saling terhubung. *Atomic design* terdiri dari lima tahapan berbeda yang bekerjasama untuk membentuk *design system* dengan cara *hierarkis* dan *sistematis* [2]. Metodologi *atomic design* terbagi menjadi lima tahapan seperti pada gambar dibawah ini.



**Gambar 2-10 Atomic Design Methodology**

### 1. Atom

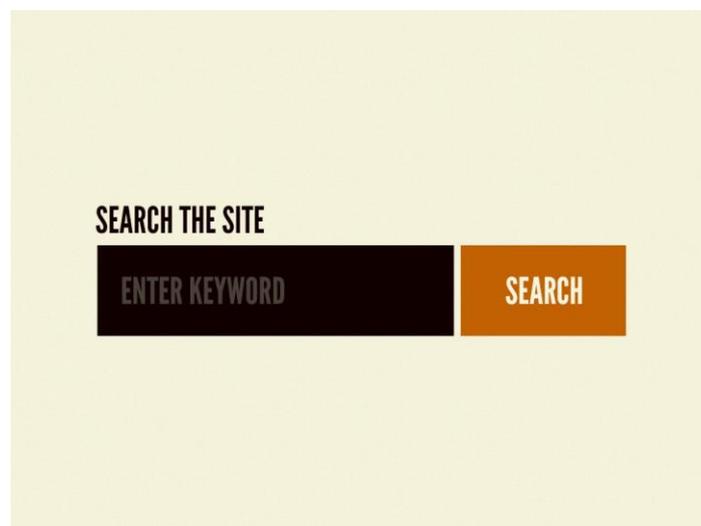
*Atom* merupakan komponen terkecil dari suatu aplikasi, yang berfungsi sebagai blok bangunan dasar yang terdiri dari komponen element seperti *label*, *form*, *inputfield*, *button* dan komponen lainnya yang tidak bisa lagi dipecah menjadi bagain terkecil [2].



**Gambar 2-11 Tahapan Atom**

## 2. Molecules

*Molecules* merupakan gabungan dari pada *atom-atom*, yang berfungsi sebagai satu unit yang memiliki fungsi yang spesifik. [2].



**Gambar 2-12 Tahapan Molecules**

## 3. Organisms

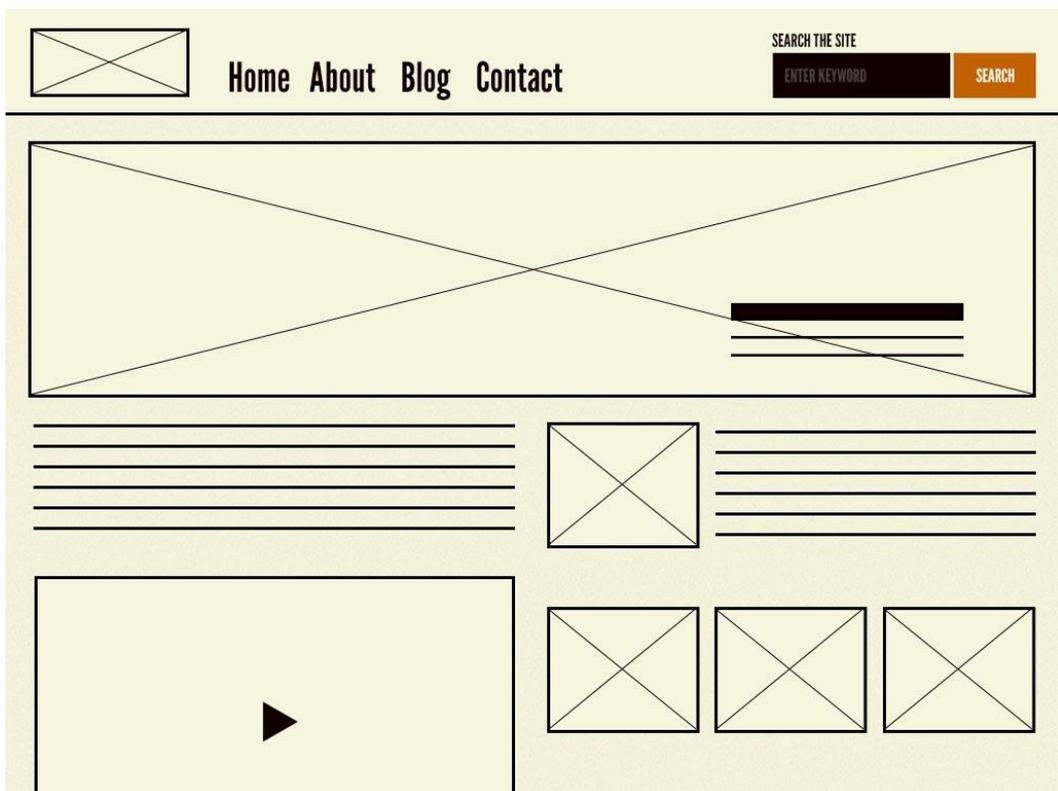
*Organism* merupakan komponen *user interface* yang relative kompleks, terdiri dari gabungan *molecules* dan *atom* [2]. *Organism* ini membentuk bagian yang berbeda dari suatu antarmuka.



**Gambar 2-13 Tahapan Organisms**

#### 4. Templates

*Templates* merupakan *page-level object* yang menempatkan komponen kedalam halaman *layout* dan mengartikulasikan struktur konten yang mendasari desain antarmuka. Dengan kata lain, adalah kerangka dasar dari kumpulan *organisms* [2].

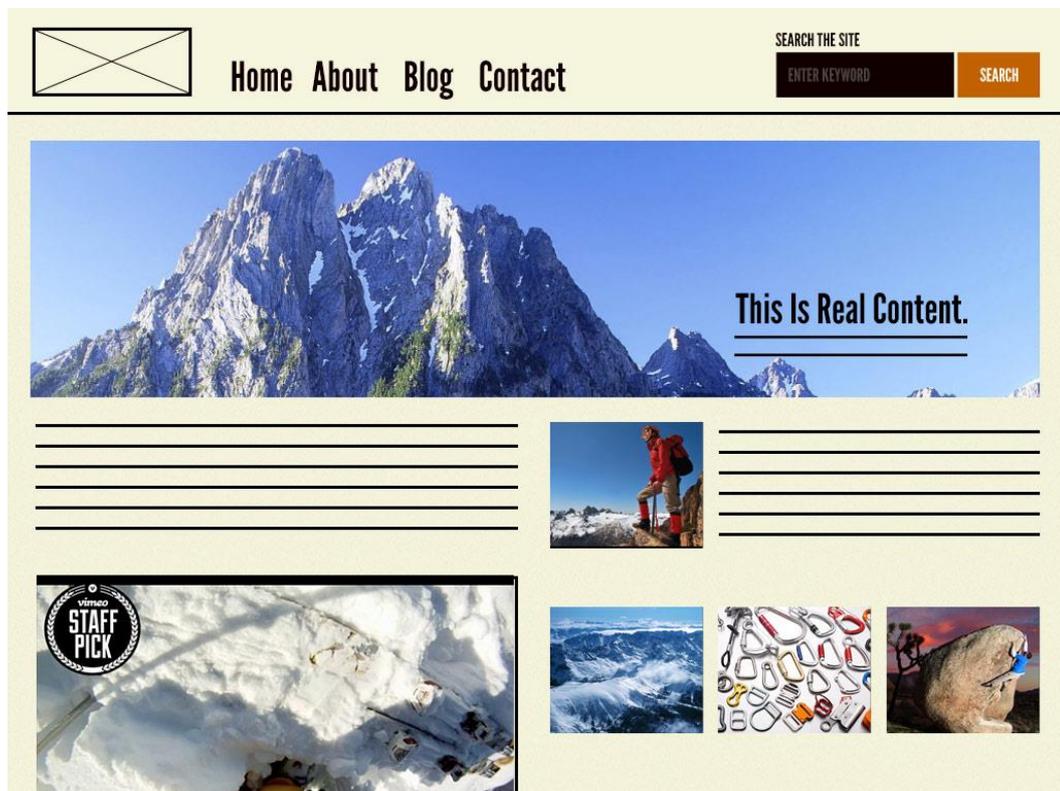


**Gambar 2-14 Tahapan Templates**

#### 5. Pages

*Pages* merupakan bentuk implementasi akhir dari kumpulan *templates*. *Pages* ini menampilkan seperti apa tampilan *user interfaces* dengan konten *representative*

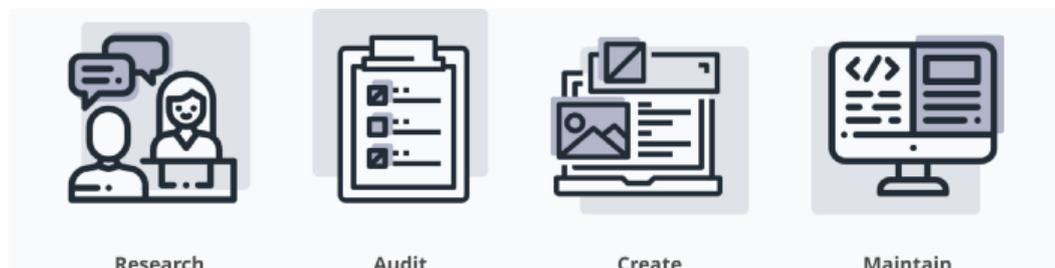
nyata. *Pages* juga sangat penting karena ini nantinya akan dilihat oleh *end user*, dan bagaimana *user* nanti akan berinteraksi [2].



Gambar 2- 15 Tahapan pages

## 2.7 Strategizing a Roadmap

Setelah tim produk terbentuk dan tujuan telah ditetapkan, maka *roadmap* pun diperlukan untuk merencanakan program kerja kedepannya. *Roadmap* adalah suatu peta ataupun panduan yang bisa digunakan sebagai petunjuk dalam strategi perencanaan. Terdapat 4 element penting *roadmap design system* diantaranya: *research*, *audit*, *create* dan *maintain* [16].



Gambar 2-16 Strategizing a Roadmap Design System

## 2.8 UX Research

*User experience (UX) Research* adalah proses penelitian sistematis yang ditujukan untuk memahami kebutuhan, tingkah laku, sikap hingga pain points pengguna saat menggunakan produk *digital* dan fisik, beserta pengalaman yang menyertainya [17]. *UX research* memiliki cakupan metode penelitian yang cukup banyak, sehingga bisa mendapatkan banyak informasi mengenai kebutuhan maupun pain points dari para penggunanya. Tujuan utama dari *UX research* sendiri yaitu untuk memberikan desain produk yang sesuai dengan perspektif pengguna.

### 2.8.1 Penelitian Kualitatif

Tujuan dari penelitian kualitatif untuk mengetahui sikap, kebutuhan, maupun motivasi pengguna. Metode yang bisa digunakan pun cukup banyak seperti wawancara dan studi lapangan etnografi [17]. Dari penelitian kualitatif akan mendapatkan pemahaman mendalam mengapa pengguna melakukan apa yang mereka lakukan. Misalnya, melakukan wawancara dengan sejumlah kecil pengguna dan mengajukan pertanyaan terbuka untuk mendapatkan wawasan pribadi tentang kebiasaan olahraga mereka, kebiasaan apa yang mereka lakukan dll.

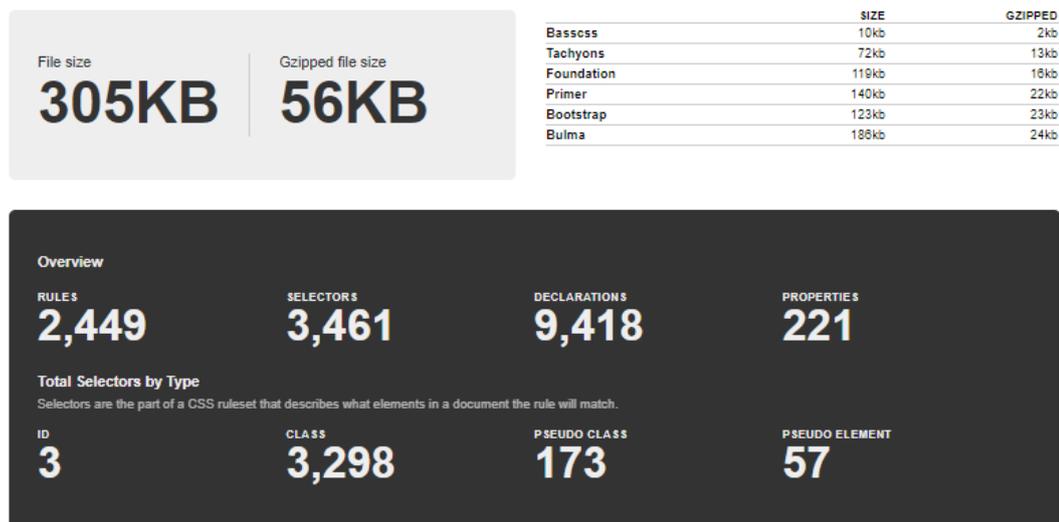
### 2.8.2 Penelitian Kuantitatif

Tujuan penelitian kuantitatif untuk menguji hasil produk berdasarkan pengalaman pengguna. Metode penelitian kuantitatif lebih terstruktur (seperti survei dan analitik). Pengumpulan data kuantitatif harus terukur misalnya melakukan survei *online* kepada pengguna untuk menjawab pertanyaan tentang kebiasaan olahraga mereka : “Berapa jam anda berolahraga dalam satu minggu?”. dengan data ini dapat menemukan pola di antara grup pengguna. Pengumpulan data kuantitatif harus secara objektif tanpa adan kehadiran, kepribadian atau asumsi [17].

## 2.9 Audit Design

Secara istilah Audit merupakan pengumpulan dan pemeriksaan terkait suatu informasi [18]. Setelah proses *research*, dilakukanlah proses *conducting a visual audit* terhadap tampilan sistem yang sudah ditentukan, dan mendokumentasikannya kedalam bentuk *interface inventory* [9]. Tujuan dari *audit design* ini untuk

mengetahui dan mengelompokkan berapa banyak aturan, *element*, *property* dan *stylesheet* yang dimiliki oleh *system*. Terdapat *tool* yang bisa digunakan untuk melihat seberapa banyak aturan, *element*, *property* maupun *stylesheet* pada suatu website, yaitu *cssStats*. Hanya saja tidak semua website bisa dilihat seberapa banyak aturan tersebut menggunakan *CSS Stats*.



#### Total Declaration Counts

A declaration represents a property value pair. e.g. display: block would represent 1 declaration

#### LAYOUT AND STRUCTURE

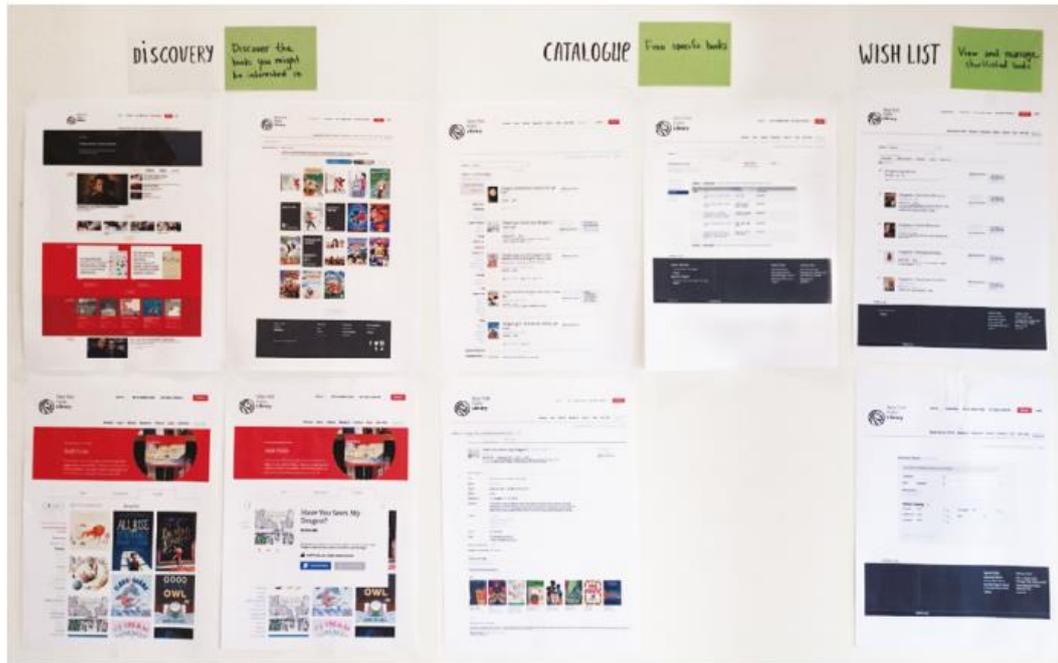
Display	Float	Width	Height	Max Width
<b>546</b>	<b>4</b>	<b>480</b>	<b>481</b>	<b>46</b>
Min Width	Max Height	Min Height		
<b>34</b>	<b>32</b>	<b>18</b>		

**Gambar 2-17** *CssStats* pada *Instagram.com*

Ada 2 Tahapan yang dilakukan ketika Proses *conducting a visual audit* yaitu [4]:

### 2.9.1 Identify Key Behaviors

*Identifikasi key behaviors* adalah suatu rangkaian aktivitas yang bertujuan untuk mengidentifikasi kebutuhan utama dan perilaku pengguna yang dilakukan di setiap *segmen journey* pengguna terhadap penggunaan produk [4]. Adapun proses yang dilakukannya adalah mengidentifikasi *behavior* suatu halaman informasi pada website maupun aplikasi, dan dibuat *labeling journey* sesuai fungsi halaman informasi tersebut.



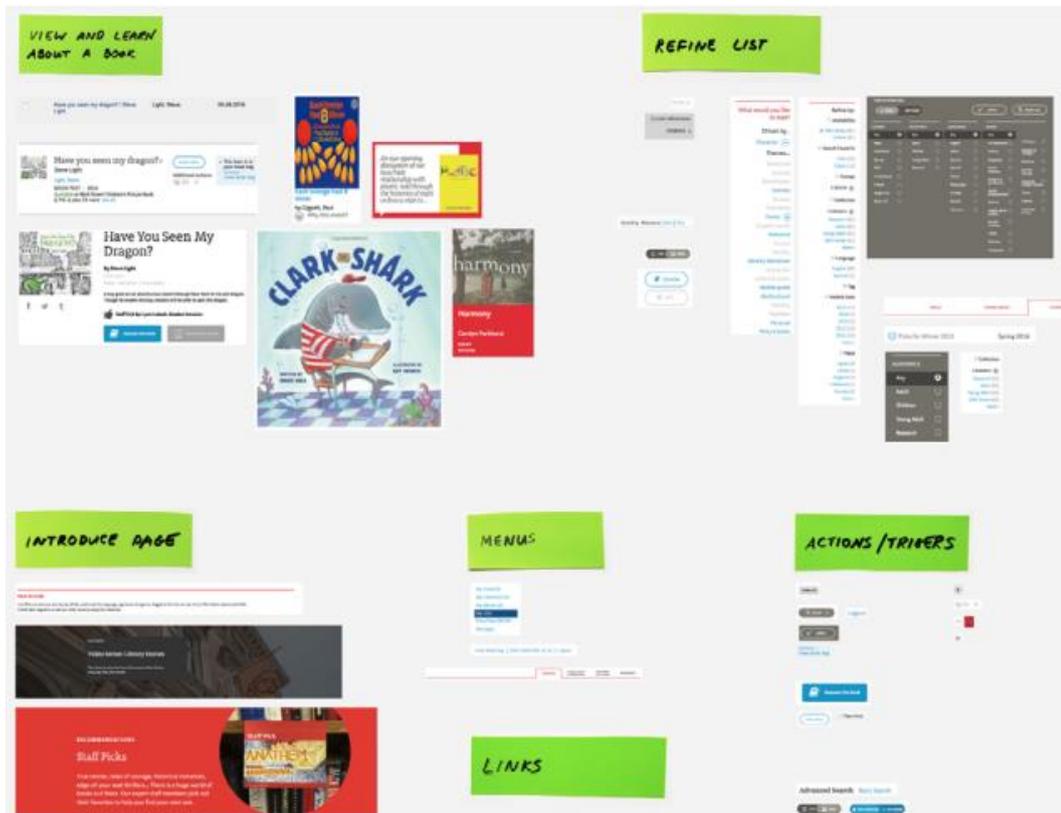
**Gambar 2-18 Proses Identify Key Behaviors**

Adapun *labeling behaviour* yang dapat digunakan adalah sebagai berikut [4]:

**Tabel 2-1 Label Identifikasi Key Behaviour**

Label	Behaviour
Discovery	Tingkah laku pengguna disaat membuka halaman aplikasi adalah memindai dan menelusuri informasi yang diinginkan.
Information	Tingkah laku pengguna yang dilakukan adalah menerima informasi dari suatu halaman aplikasi.
Catalog	Tingkah laku pengguna yang dilakukan adalah melihat list suatu informasi dan melihat informasi tersebut
Finding	Tingkah laku pengguna yang dilakukan adalah mencari informasi tertentu dalam suatu halaman aplikasi.
Managing	Tingkah laku pengguna yang dilakukan adalah mengelola informasi menambah, pengubah, dan menghapus informasi.





**Gambar 2-20 Proses break behaviors into action**

## 2. Define Pattern

*Define Pattern* adalah tahapan menentukan *elements group* dan memutuskan apakah *components* yang ini digabungkan menjadi satu pola yang sama atau terpisah. Dalam menentukan pola pada *component* ada dua tahapan yang bisa dilakukan yaitu : *specificity scale* dan pemetakan *content structure* [4].

### a. Specificity Scale

*Specificity Scale* adalah suatu proses atau tahapan penentuan *components* yang bertujuan untuk menentukan apakah *components* ini *specific* atau *generic* atau apakah *components* ini termasuk *reusable components* atau *components* khusus [4].

### b. Content Structure

*Content Structure* adalah suatu proses atau tahapan penentuan hirarki *element* dan menentukan bagaimana pengelompokan suatu modul *components*. Dimana setiap struktur hirarki pada modul itu diuraikan dan ditentukan setiap *element* yang ada pada *components*. Hal ini bertujuan untuk memberikan pemahaman yang sama dalam melihat suatu modul *components* antara desainer dan *developer* [4].

c. Naming

*Naming* adalah proses penamaan pada suatu modul *components*, hal ini sangatlah penting dilakukan agar setiap orang dalam tim memiliki pemahaman yang sama dalam melihat *components* [4].

d. Visual Hierarchy

*Visual Hierarchy* adalah proses pengurutan *element*, dimana pengguna akan memproses informasi pada suatu halaman. Fungsi visual hierarchy adalah untuk memungkinkan pengguna memahami informasi dengan lebih mudah ketika melihat *user interface* [4].

## 2.11 Perceptual Pattern

*Design system* terdiri dari berbagai *element* baik yang *independent* atau berdiri sendiri namun meskipun *independent element* tersebut saling terikat dan saling mempengaruhi satu sama yang lainnya. *Design system* dibagi menjadi dua bagian yaitu *perceptual pattern* dan *functional pattern* [4].

*Perceptual Pattern* merupakan kumpulan suatu pola yang berperan sebagai aturan dasar atribut *style* yang dimiliki oleh suatu *system* atau produk [4]. *Perceptual pattern* mencakup lapisan *foundation* dengan membuat aturan visual dari *element* dasar seperti *color, palette, typography, iconography, spacing & layout* serta ilustrasi. *Style guidelines* juga bagian dari cakupan *perceptual pattern*. Yang mana *style guideline* merupakan kumpulan panduan desainer dalam perancangan tampilan antarmuka atau *user interface* [13]. Harapan adanya *style guideline* adalah untuk meningkatkan konsistensi setiap halaman maupun di berbagai aplikasi [13].