

## **BAB 2**

### **TINJAUAN PUSTAKA**

#### **2.1 Profil PT. Codepolitan**

PT. Codepolitan integrasi Indonesia atau Codepolitan adalah sebuah perusahaan yang bergerak di bidang edukasi dan teknologi. Berdiri sejak Februari 2016 dan berkedudukan di kota Bandung. Codepolitan memiliki layanan seperti online course, coding Bootcamp, Corporate Training, Media dan Tech Adoption. Online course codepolitan merupakan program belajar coding secara online dengan kurikulum berstandar industri yang memiliki keunggulan alur belajar terarah, waktu belajar fleksibel, progress belajar terukur, kurikulum standar industri, video tutorial, sertifikasi online, garansi uang Kembali, dan lain sebagainya. Coding Bootcamp codepolitan merupakan program pelatihan coding secara intensif dengan bimbingan mentor professional. Media & Tech Adoption yaitu membantu technology owner mempromosikan teknologinya kepada developer Indonesia. Codepolitan juga mempunyai campaign media seperti online course, live webinar, Event, advertorial, Challenge

#### **2.2 Sejarah Singkat Codepolitan**

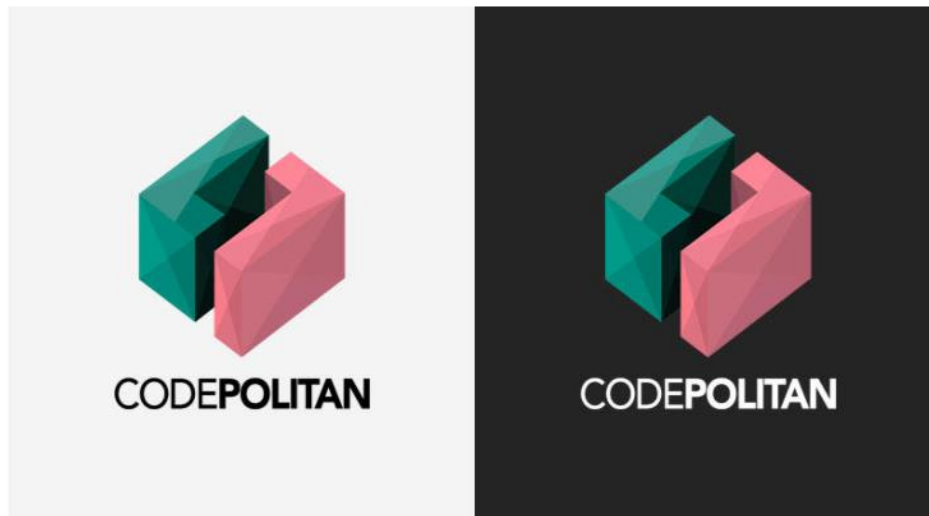
Pada tahun 2016 Codepolitan memulai segala sesuatunya dengan portal media online yang membahas tentang pemrograman dan teknologi, lalu lahirnya program coding bootcamp, Codepolitan Developer School. lalu menyelenggarakan salah satu event terbesar programmer di Indonesia, Indonesia Developer Summit pada tahun 2017. Lalu pada tahun 2018 diluncurkannya program Kelas Online Premium Membership, dengan 1000 purchase di bulan pertama peluncuran, pada tahun 2019

peluncuran program kelas online lifetime dan menjadi salah satu perusahaan terdepan dan konsisten yang memberikan pembelajaran dalam bidang programming di Indonesia. Pada tahun 2020 sampai hari ini Codepolitan menjadi salah satu perusahaan yang bertahan di masa pandemi dengan inovasi dalam program kelas online, peluncuran program kampus coding dan Geekmentor.

### **2.3 Visi dan Misi Codepolitan**

- **Visi**  
Menjadi komunitas developer terbesar di Indonesia
- **Misi**  
Menjadi pusat informasi bagi developer di Indonesia mengedukasi market untuk melahirkan banyak developer berkualitas meningkatkan skill developer agar selaras dengan kebutuhan industry mewadahi kreativitas developer dalam berkarya

### **2.4 Logo Codepolitan**



Gambar 2. 1 Logo Perusahaan

Berikut ini makna dari logo baru CodePolitan:

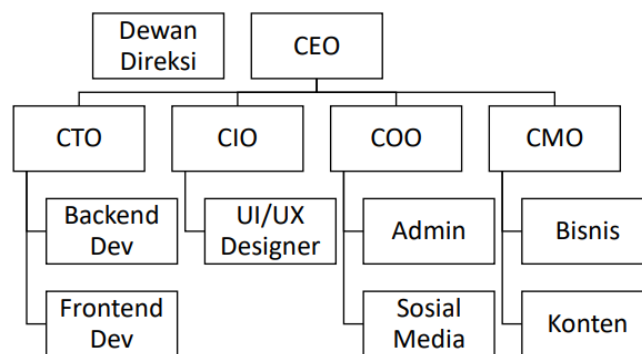
Bentuk logo menyerupai tanda kurung "<" dan ">" (*bracket*) menunjukkan fokus codepolitan sebagai media edukasi dan informasi yang membahas seputar pemrograman. Karena tanda kurung adalah sebuah tanda yang sangat familiar sekali dalam pemrograman. Bentuk logo yang menyerupai tanda kurung (*bracket*) berbentuk balok bata yang saling terhubung menggambarkan semangat kolaborasi dalam membangun ekosistem pemrograman di Indonesia.

Warna logo terdiri dari hijau toska (#1FBBA6) dan magenta (#F16767) menunjukkan identitas kami sebagai CodePolitan. Warna hijau toska (#1FBBA6) menggambarkan keinginan untuk belajar. Warna hijau toska merupakan warna yang terbentuk karena percampuran antara warna hijau dan biru. Itu menggambarkan bahwa dalam belajar kita harus senantiasa merasa muda (hijau) sehingga bisa membuka diri untuk menerima informasi dan belajar ilmu baru dalam transisi untuk menjadi profesional (biru). Warna magenta (#F16767) menggambarkan semangat

yang konsisten dalam belajar. Dalam belajar, teratur atau konsisten jauh lebih baik dari pada menggebu-gebu namun tidak berkelanjutan. Sehingga untuk menggambarkan semangat (merah) dan keteraturan (biru), kami menggunakan magenta (bukan pink) yang merupakan pencampuran antara warna merah dan biru.

Tulisan CODEPOLITAN menunjukkan identitas kami sebagai CodePolitan. Kata **CODE** ditulis lebih tipis dari kata **POLITAN** bermakna proses. Untuk menjadi seorang yang ekspert kita harus siap memulainya dari newbie. Untuk menjadi besar kita harus mulai dari yang kecil. Dalam menggapai sesuatu kita harus siap menjalani dan menikmati semua prosesnya. Sehingga ketika kita telah sampai pada apa yang kita ingin gapai kita tidak lantas menjadi sombong, dan ketika kita belum bisa menggapainya kita tidak akan menyerah dalam melangkah. Tulisan CODEPOLITAN pada logo, ditulis dengan font *Avenir* yaitu sebuah font berjenis sans serif yang dibuat oleh *Adrian Frutiger*.

## 2.5 Struktur Organisasi



Gambar 2. 2Struktur Organisasi

## 2.6 Landasan Teori

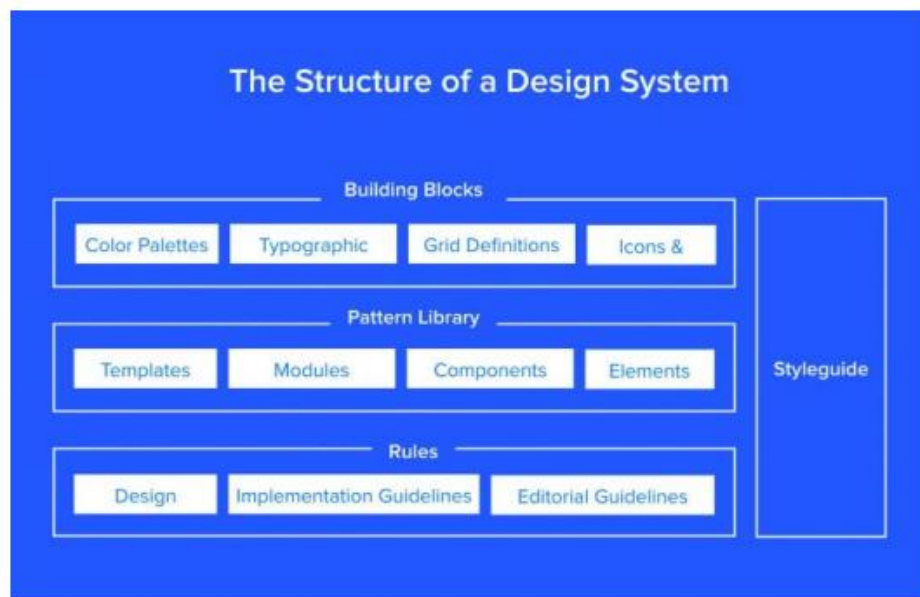
Penelitian dilakukan berlandaskan dengan berbagai teori – teori yang dijadikan acuan. Berikut teori - teori pendukung yang akan digunakan dalam penelitian yaitu diantara lain:

### 2.6.1 *Design System*

*Design System* merupakan sebuah sekumpulan komponen-komponen yang bisa digunakan kembali sehingga dapat dipandu dengan standar yang jelas, dan juga dapat dirakit bersama tim atau organisasi untuk membangun beberapa aplikasi[1] . Jadi *Design System* dapat membantu tim dalam membangun produk yang bisa lebih baik dan lebih cepat, dan dengan membuat desain yang dapat digunakan kembali penggunaan ulang memungkinkan penskalaan.dan ini merupakan inti dan juga nilai utama *design system*[1]. *Design system* menyediakan dokumentasi yang meliputi *Styleguide, code library, pattern Library*.

*Design System* dapat meliputi dokumentasi yang juga mewakili design principle, design language , pattern library, pengkodean dan serta brand identity[4]. Tujuan pada *design system* yaitu untuk mengelola elemen desain visual untuk di terapkan oleh perusahaan, yang dimana mencegah inkonsistensi, dan untuk memastikan bahwa komponen perusahaan anda dapat terlihat sama dimanapun komponen itu muncul[1].

Struktur pada *design system* juga terdiri dari empat subclass hirarki yaitu yang pertama *Building Block* terdiri dari elemen foundations seperti (Color Palettes, typographic, Grid dan Icon), yang kedua yaitu Pattern Library yang terdiri dari sekumpulan komponen, element dan lain sebagainya, yang ketiga Rules yang terdiri pedoman standar dan panduan penggunaan, yang terakhir yaitu *Styleguide*[5].



Gambar 2. 3 Struktur design system

## 2.6.2 Komponen

Komponen merupakan bagian dari *code* yang dapat digunakan kembali dalam sistem dan dapat berfungsi sebagai blok pembangun antarmuka sistem. Komponen juga berkisar dalam kompleksitas. Misalnya contoh: dengan mengurangi komponen menjadi satu fungsi, seperti tombol untuk meningkatkan fleksibilitas, akan membuatnya lebih dapat digunakan kembali.

Contoh komponen yang lebih kompleks, contohnya yaitu seperti jika ada tabel untuk tipe data tertentu yang dapat melayani kasus penggunaannya dengan baik, tetapi kompleksitas ini membatasi jumlah skenario yang berlaku. Semakin banyak komponen yang dapat digunakan kembali, semakin sedikit yang perlu Anda pelihara, dan semakin mudah skalanya[1].

## 2.6.3 Standards

Dengan memahami bukan hanya apa, tetapi mengapa, di balik *Design System* itu sangat penting untuk menciptakan pengalaman pengguna. Mendefinisikan dan mematuhi standar adalah bagaimana kita menciptakan pemahaman tersebut. Dengan menciptakan pemahaman tersebut kita akan menghilangkan subjektivitas dan ambiguitas yang sering menciptakan gesekan dan kebingungan yang ada di dalam tim produk. Standar mencakup desain dan pengembangan. Menstandarisasi hal-hal seperti konvensi penamaan, persyaratan aksesibilitas, dan struktur file akan membantu tim bekerja konsisten dan mencegah kesalahan. Tanpa standar, keputusan menjadi sewenang-wenang dan sulit untuk mengkritik. tidak hanya tidak menskala, tetapi juga menciptakan pengalaman pengguna yang tidak konsisten. Standar mengatur tujuan, gaya, dan penggunaan komponen[1].

Bahasa visual adalah bagian inti dari standar desain yang dibuat. Dengan menentukan tujuan dan gaya warna, bentuk, jenis, ikon, ruang, dan gerakan yang sangat penting untuk menciptakan pengalaman pengguna yang selaras dan konsisten dengan merek. Setiap komponen dalam sistem menggabungkan elemen-elemen ini, dan memainkan peran integral dalam mengekspresikan kepribadian merek[1].

#### **2.6.4 Perceptual Pattern dan Functional Pattern**

*Perceptual pattern* merupakan sesuatu gaya abstrak dari sebuah produk yang dapat membantu produk agar lebih diingat oleh pengguna lewat cara yang mengekspresikan dan mengkomunikasikan maksud dan makna dari produk secara *visual*. *Perceptual Pattern* juga mencakup elemen abstrak seperti *pallette* warna, tipografi, spasi, *layout*, ilustrasi, icon, maupun animasi[6].

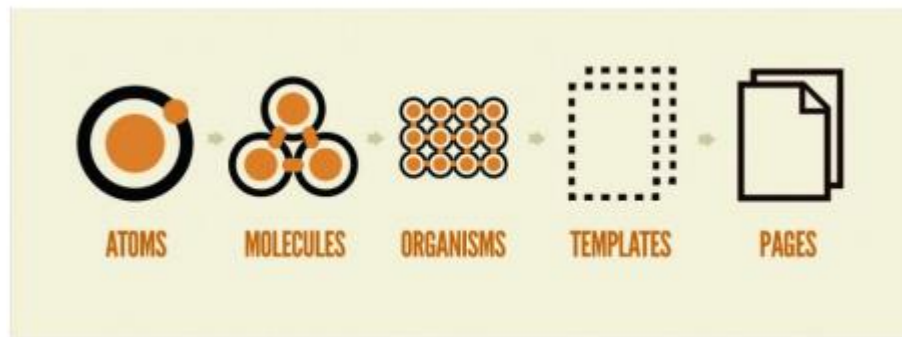
*Functional Pattern* adalah suatu modul yang konkrit dari sebuah antarmuka yang mana terdiri dari kumpulan elemen-elemen komponen yaitu seperti dialog dan 9

element ,button, navigasi, dan lain sebagainya yang dapat mendorong perilaku dan juga aksi pengguna untuk dapat melakukan aksi pada dalam kondisi tertentu[7].

### 2.6.5 Atomic Design

*Atomic Design* adalah sebuah metode untuk pendekatan desain yang diperkenalkan oleh Brad Forst, *atomic design* juga bekerja dengan cara memecah semua komponen pada sebuah tampilan antarmuka menjadi komponen-komponen dasar yang hierarkis, sistematis dan dapat saling terhubung. Karena dengan cara kerjanya *atomic design* mampu dimanfaatkan untuk mempercepat sebuah proses pembangunan modular desain[8].

Terdapat lima tahapan yang ada pada metode *atomic design*, yaitu sebagai berikut :

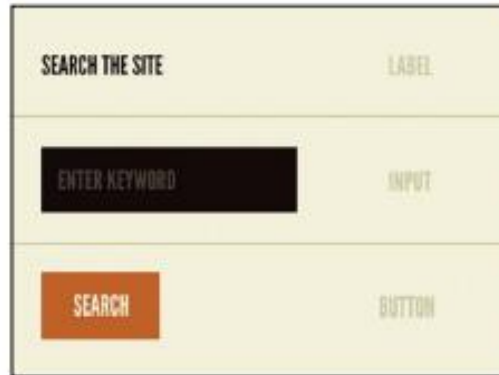


Gambar 2. 4 Tahapan Atomic Design

#### 1. *Atoms*

*Atoms* merupakan sebuah bahan penyusun materi dasar antarmuka perusahaan sebagai fondasi dasar yang terdiri dari antarmuka pengguna yang dapat dipecah lebih jauh tanpa berhenti berfungsi contohnya: HTML, *element* dasar, label formular, input, tombol, dan lain sebagainya[7].





Atoms include basic HTML tags like inputs, labels, and buttons.

Gambar 2. 5 Atom

## 2. Molecules

*Molecules* adalah kelompok UI yang *relative* sederhana elemennya yang berfungsi bersama sebagai satu kesatuan, contohnya: Tabel formular, pencarian, input, dan *button* atau tombol, yang dimana dapat bergabung bersama untuk membuat pencarian dengan membentuk sebuah molekul[7].

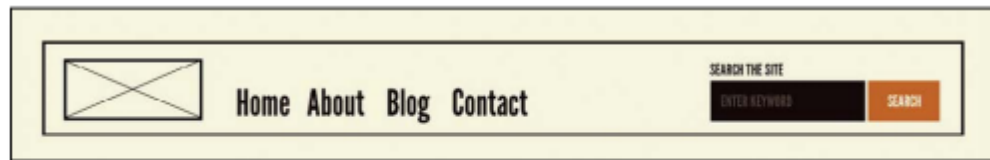


A search form molecule is composed of a label atom, input atom, and button atom.

Gambar 2. 6 Molekul

## 3. Organisme

*Organisme* adalah kumpulan komponen ui yang *relative* kompleks yang terdiri dari kelompok molekul atau atom atau *organisme* lain, *organisme* juga dapat terdiri dari jenis molekul yang sama ataupun yang berbeda, contoh *element* yang berbeda yaitu gambar, logo, daftar navigasi utama dan formular pencarian[7].

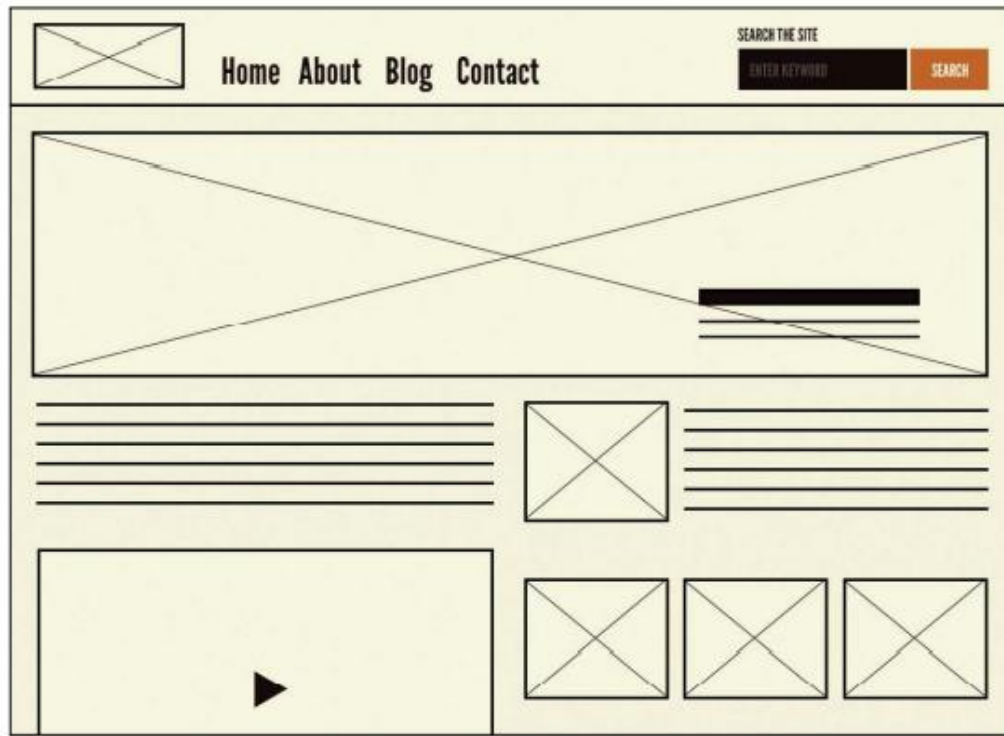


This header organism is composed of a search form molecule, logo atom, and primary navigation molecule.

Gambar 2. 7 Organisme

#### 4. *Template*

*Template* adalah objek tingkat halaman yang menempatkan komponen ke dalam tata letak dan mengartikulasikan struktur konten yang mendasari desain, contohnya kita dapat mengambil organisme header dan menerapkannya ke halaman beranda[7].

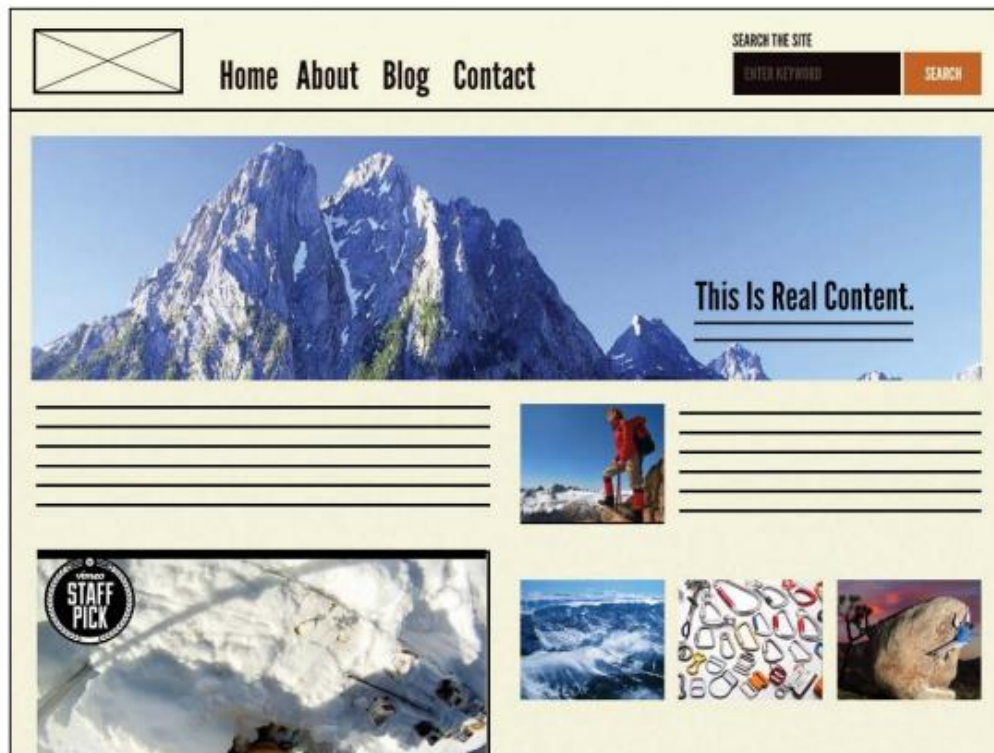


The homepage template consists of organisms and molecules applied to a layout.

Gambar 2. 8 Template

## 5. Pages

*Pages* adalah contoh khusus dari template yang menentukan apa itu Ui, sebagai contoh kita dapat mengambil *template* beranda dan menuangkannya teks, gambar, dan media representatif ke dalam template untuk ditampilkan sebagai konten nyata dalam sebuah aksi[7].



The page stage replaces placeholder content with real representative content to bring the design system to life.

Gambar 2. 9 Pages

### 2.6.6 Brand Identity

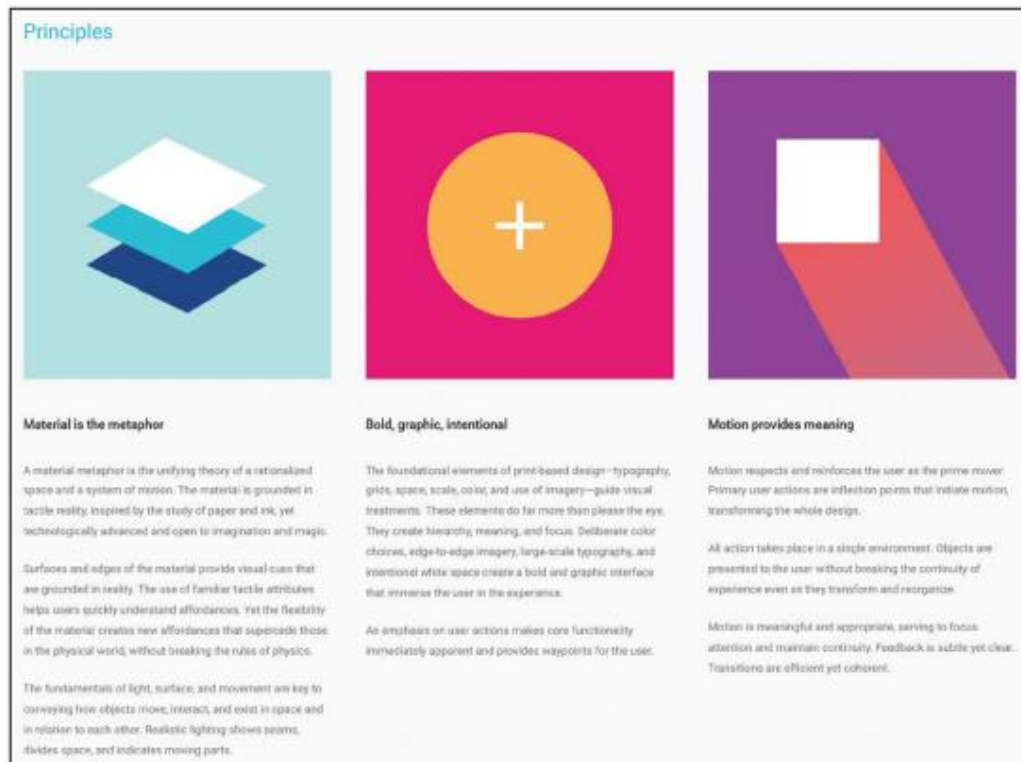
*Brand Identity* merupakan asosiasi merek yang mendefinisikan assets dan materials yang membuat perusahaan itu unik dari yang lain. Mulai dari logo, typography, color palette, slogan, dan masih banyak lagi contoh yang dapat menggambarkan dari *brand identity*[7]. Dengan brand identity dapat membangun merek yang mudah di ingat oleh konsumen dan ini semua tentang konsistensi. Kita mengingat merek-merek terbaik karena kehadirannya dengan pengulangan logo, gambar, warna, dan font yang sama. Jadi dengan hanya cukup melihat kita dapat mengenali brand tersebut.



Gambar 2. 10 Brand Identity

### 2.6.7 Design Language

*Design language* merupakan bahasa desain yang mengartikulasikan ke arah desain umum, filosofi, dan pendekatan untuk proyek atau produk tertentu. menampilkan dengan cara yang kohesif di berbagai macam produk dan media. Google mengembangkan bahasa desain yang disebut *material design*. Panduan dari gaya material design mendefinisikan filosofi desain menyeluruh, tujuan, dan prinsip umum, sementara juga menyediakan aplikasi spesifik dari bahasa material design[7].

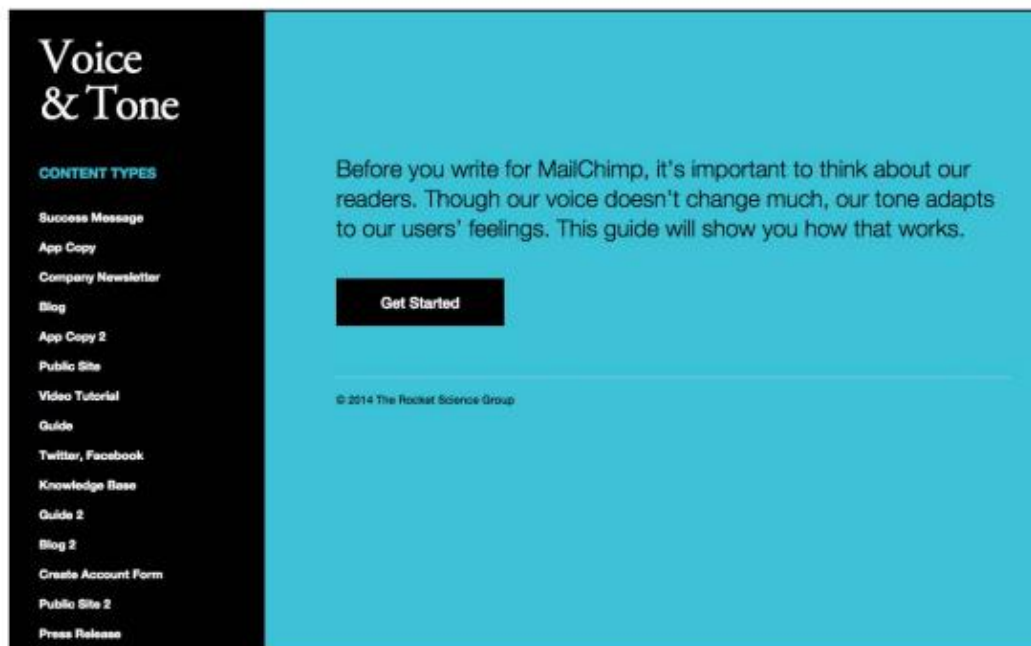


Google's material design language.

Gambar 2. 11 Material Design

### 2.6.8 Voice and tone

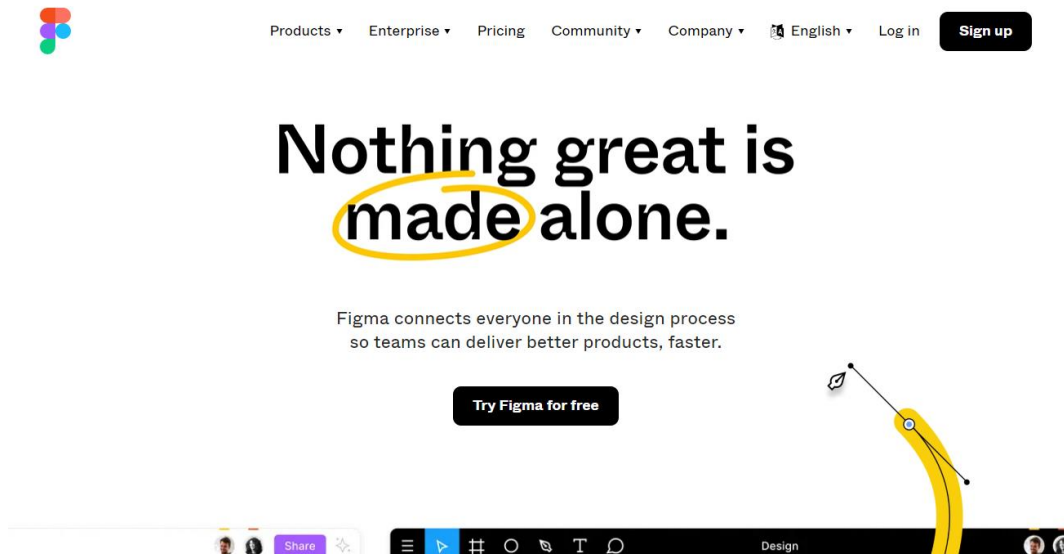
Orang-orang berinteraksi dengan merek atau brand di berbagai channel dan media. Selain media digital, merek beroperasi di media cetak, ritel, radio, TV, dan lainnya sebagainya. Ketika sebuah merek diharuskan berkomunikasi dengan begitu banyak variasi titik kontak, berbicara dengan cara yang seragam dan konsisten menjadi hal yang sangat penting bagi kesuksesan merek itu sendiri. Voice merupakan aspek elemen dari identitas merek, jadi biasanya merek pedoman identitas mencakup beberapa referensi ke voice merek itu sendiri. Namun, pedoman ini biasanya tidak terlalu bernuansa, yaitu mengapa pedoman voice and tone sangat penting[7].



Gambar 2. 12 Voice and Tone

## 2.6.9 Figma

Figma merupakan aplikasi desain antarmuka yang berjalan di browser tetapi ada pula versi desktop untuk windows dan Mac OS. Figma memberikan semua tools yang dibutuhkan oleh designer untuk fase desain proyek, termasuk alat vektor yang mampu membuat ilustrasi lengkap, serta kemampuan pembuatan prototipe, dan keuntungan lainnya Figma juga memungkinkan untuk membangun *library* komponen yang dapat digunakan kembali yang dapat diakses oleh seluruh tim. Komponen memberi desainer permulaan pada sistem desain yang ada, dan ketika komponen tersebut diperbarui di *library* utama, maka perubahan itu dilakukan pada semua desain untuk seluruh tim[8].



Gambar 2. 13 Figma

### 2.6.10 Team Model

Team model menyatukan orang-orang yang sama pentingnya untuk dimasukkan ke tim dalam pembuatan design system. veteran design system yaitu Nathan curtis menguraikan adanya tiga model tim populer yang digunakan di oleh beberapa perusahaan, diantara lain sebagai berikut[1].

1. **The solitary model**

Model soliter”team” yaitu yang mengatur design system hanya satu orang untuk mengatur semuanya.





*Gambar 2. 14 The Solitary Model*

## **2. The centralized team model**

Sistem model ini yaitu satu team mempertahankan design system sebagai pekerjaan dengan penuh waktunya mereka seperti satu team mengatur sistem.



*Gambar 2. 15 The Centralized Model*

## **3. The federated model**

Anggota team dari seluruh perusahaan berkumpul untuk bekerja pada sistem. Model design system gabungan ini contohnya seperti orang-orang dari berbagai tim berkumpul untuk mengatur dan mengelola sistemnya.



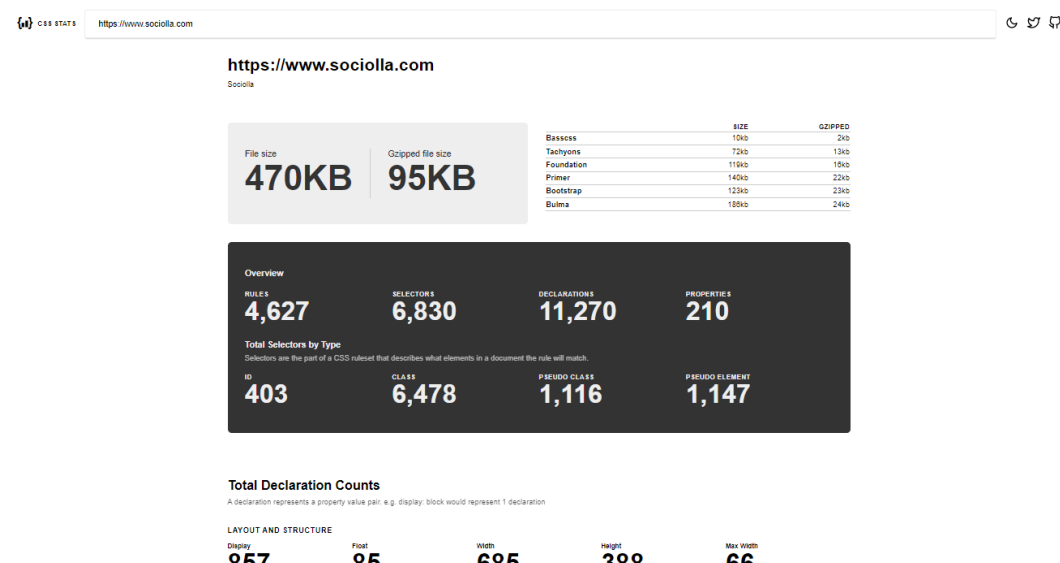
*Gambar 2. 16 The Federated Model*

### **2.6.11 In Depth Interview**

In-depth interviews merupakan salah satu metode pengumpulan data primer yang paling efisien. Dilakukan dengan mengungkap detail secara mendalam dari pengalaman orang yang akan di wawancarai atau perspektif pada suatu topik. Keuntungan dari in depth interviews ini dapat membantu mengungkap lebih banyak informasi rinci secara mendalam dibandingkan menggunakan metode pengumpulan data lainnya contohnya survey. In depth interviews juga dapat digunakan untuk mengeksplorasi konsep lebih lanjut seperti investigasi dan analisis deskriptif. Pembicara in-depth interviews juga dapat mengungkapkan isi pikiran mereka secara essai yang kasusnya pun bukan dalam skala penilaian atau kuesioner[9].

### **2.6.12 Audit Design**

Audit desain melibatkan analisis semua elemen desain yang digunakan di seluruh website untuk memastikan bahwa *branding* konsisten[10]. Audit design dilakukan dengan cara mengekstraksi style desain seperti properti, elemen visual, dan identitas yang dimiliki oleh perusahaan. Menganalisis elemen visual yang ada pada produk dapat mempermudah designer dalam memperbaiki ketidakkonsistenan. Adapun salah satu tools yang digunakan untuk mengekstraksi style desain adalah CssStats. Tools ini dapat menampilkan berapa banyak warna unik, keluarga font, dan ukuran font yang dimiliki oleh website tersebut.

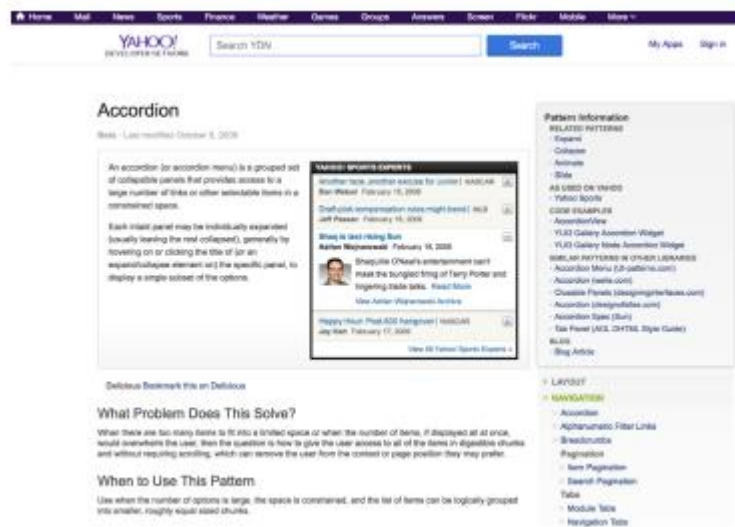


Gambar 2. 17 Tools CssStats

### 2.6.13 Pattern Library

Sebuah design system tidak hanya terdiri dari pola, tetapi juga terdapat teknik dan praktik untuk membuat, menangkap, berbagi, dan mengembangkan pola-pola tersebut. Pattern library merupakan alat untuk menangkap, mengumpulkan, dan berbagi prinsip, design pattern, dan guideline tentang tata cara penggunaannya.

Secara tradisional, pattern library ini berfokus pada style, seperti icon, warna, dan tipografi, sedangkan pattern library mencakup seperangkat pola yang lebih luas lagi. Pattern library terkadang dianggap dapat dipertukarkan dengan design system. Tetapi bahkan yang paling komprehensif dan pattern library hidup bukanlah sebagai sistem itu sendiri tetapi merupakan sebagai alat yang membantu membuat design system lebih efektif[11].



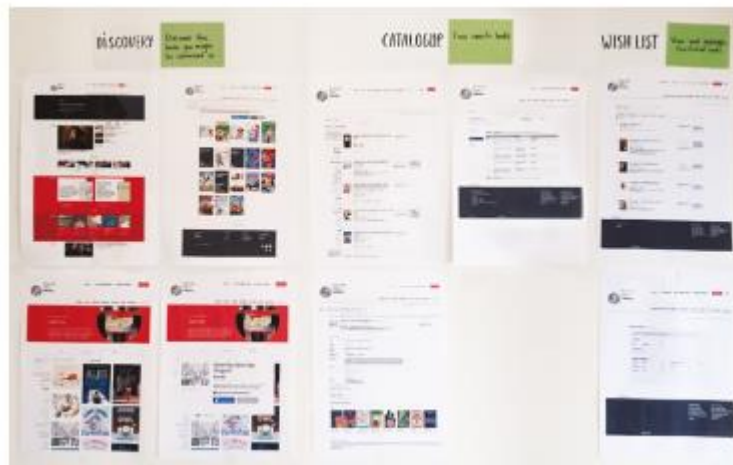
Gambar 2. 18 Pattern Library

## 2.6.14 Style Guide

Style guide merupakan suatu dokumen yang berisi seperangkat aturan atau standar untuk warna, gaya grafis, tipografi, icon, gambar dan prinsip – prinsip brand yang memberikan arahan tentang tata cara menggunakannya[12]. Keuntungan dari style guide membantu konsistensi visual pada seluruh produk dan merek perusahaan. Menciptakan hubungan yang saling percaya dengan konsumen atau pengguna. Kemudian juga membantu memudahkan designer atau anggota tim baru untuk dapat dengan mudah berpartisipasi atau membuat keputusan desain[12].

### 2.6.15 Identifikasi Key Behaviour

Identifikasi key behaviour merupakan suatu kegiatan yang mengidentifikasi kebutuhan dan perilaku utama pengguna yang dilakukan pada setiap segmen perjalanan atau journey pengguna dalam menggunakan produk digital[11]. proses yang dilakukan pada identifikasi key behaviour dengan mengidentifikasi behaviour suatu halaman pada aplikasi ataupun website, membuat labelling dan journey dari tiap komponen yang ada pada halaman tersebut sesuai dengan fungsinya.



*Gambar 2. 19 Identifikasi Key Behaviour*

### 2.6.16 Break Behaviour Into Break

Setelah mengidentifikasi key behaviour. pada tahap ini dilakukan proses pemecahan tindakan aktivitas menjadi tindakan yang lebih spesifik yang dilakukan pada setiap bagian halaman website atau aplikasi. Guna untuk mengidentifikasi suatu

tindakan yang diulang dengan memiliki pola yang sama. Sehingga dapat diketahui bahwa elemen antarmuka yang bisa digunakan kembali atau reuseable[11].



Gambar 2. 20 Break Behaviour

### 2.6.17 Consistency

Konsistensi merupakan salah satu molekul DNA Desain. Desain yang konsisten yaitu desain yang intuitif. kegunaan dan kemampuan belajar meningkat ketika elemen serupa memiliki tampilan dan fungsi yang konsisten dengan cara yang sama. Dengan adanya konsistensi dalam desain orang lain dapat mempelajari hal-hal baru dengan cepat. Keuntungan dari konsistensi yaitu pengguna akan belajar lebih cepat bagaimana menggunakan desain yang dimiliki. Dengan elemen yang konsisten dalam desain pengguna akan selalu mengingat dengan hanya melihat bahwa itu adalah brand anda. Konsistensi menghilangkan kebingungan, menghemat uang dan waktu[13].

## 2.6.18 Usability Metrics



Gambar 2. 21 Usability

*Usability* berdasarkan (ISO 9241-11) pada tahun 1998 *usability* merupakan sesuatu yang dapat di test dengan sejauh mana sebuah produk/sistem dapat digunakan dengan baik oleh pengguna untuk mencapai tujuan tertentu dengan cara yang efektif, efisien, dan puas dalam penggunaan tertentu. Menurut standar (ISO / IEC 9126) terdapat enam faktor kualitas penilaian perangkat lunak yang ada pada gambar di atas[14].

### 1. *Functionally*

*Functionally* merupakan suatu kemampuan perangkat lunak dalam menyediakan fungsi yang dapat sesuai dengan kebutuhan penggunanya.

### 2. *Reliability*

*Reliability* merupakan suatu kemampuan perangkat lunak untuk dapat mempertahankan kinerja pada kondisi tertentu.

### **3. Usability**

*Usability* merupakan suatu kemampuan perangkat lunak untuk dapat dipahami, dipelajari, digunakan dan dapat menarik penggunaanya.

### **4. Efficiency**

*Efficiency* merupakan suatu kemampuan perangkat lunak untuk menyediakan kinerja yang tepat saling berhubungan dengan sumber daya yang digunakan.

### **5. Maintainability**

*Maintainability* merupakan suatu kemampuan perangkat lunak untuk dapat memodifikasi, memperbarui, memperbaiki dan dapat beradaptasi terhadap lingkungannya.

### **6. Portability**

*Portability* merupakan suatu kemampuan perangkat lunak dalam kegiatan transfer antar lingkungan atau juga kemampuan beradaptasi di area atau titik tertentu.

Dengan menyesuaikan hasil Pre-Test tentang waktu yang akna dibutuhkan oleh *designer* dalam membuat sebuah tampilan antarmuka, dan untuk menghitung berapa lamanya waktu untuk penyelesaian tugas yang diberikan yaitu dengan menggunakannya rumus sebagai berikut

$$Effectiveness = \frac{\text{Number of tasks completed successfully}}{\text{Total number of tasks undertaken}} \times 100\%$$

*Gambar 2. 22 effectiveness*



Lalu untuk menghitung persentase efisiensi waktu yang digunakan dapat menggunakan sebuah rumus sebagai berikut[14]

$$\text{Persentase efisiensi waktu} = 100\% - \left( \frac{\text{Realisasi Waktu (RW)}}{\text{Target Waktu (TW)}} \times 100\% \right)$$

*Gambar 2. 23 Rumus Persentase*