# BAB 2
# TINJAUAN PUSTAKA

## 2.1 Profil Tokodistributor

Tokodistributor merupakan sebuah platform marketplace dengan konsep B2B (Business to Business) dan O2O (Online to Offline) yang memudahkan semua orang untuk dapat berjualan tanpa harus mempertimbangkan tempat berjualan, modal usaha, dan dapat dilakukan oleh siapapun. Dengan pendekatan konsep bisnis modern dan tradisional, Tokodistributor memberikan peluang dan kesempatan bagi semua orang untuk dapat memulai usaha dan berbisnis dengan lebih mudah, aman, dan nyaman. Semua orang dapat mengembangkan usahanya baik dengan menjadi supplier, distributor tangan pertama, umkm, reseller, dan dropshipper dengan bantuan Tokodistributor.

Tokodistributor menjawab berbagai masalah para pengusaha baru seperti masalah produk dan modal terbatas, dengan menyediakan harga produk terbaik untuk para reseller. Menjadi solusi bagi supplier yang ingin mengembangkan bisnisnya juga menjadi tujuan tokodistributor karena reseller dan supplier adalah prioritas tokodistributor.

Seiring perkembangan zaman yang terus berubah, tokodistributor terus berinovasi dan memiliki integritas untuk menjadi platform B2B terbaik di Indonesia. Karena tokodistributor memiliki tekad, bahwa tidak hanya sekedar menjadi solusi bisnis bagi reseller dan supplier tapi juga memberikan kontribusi dalam membangun perekonomian Indonesia.

### 2.1.1 Logo Tokodistributor


**Gambar 2.1 Logo Tokodistributor**

## 2.2 Landasan Teori

Dalam landasan teori akan memaparkan teori-teori yang akan digunakan dalam penelitian yang dilakukan.

### *2.2.1 Design System*

*Design System* adalah sebuah pola *(pattern)* yang saling berhubungan satu sama lain dan digunakan untuk bersama-sama yang terorganisir secara saling terhubung untuk mencapai tujuan dari sebuah produk [6]. *Design system* memungkinkan tim untuk membuat produk yang lebih baik dan lebih cepat, dengan desain yang dapat digunakan secara berulang-ulang. *Design system* adalah kumpulan komponen yang dapat digunakan berulang kali, dengan aturan yang jelas dan dapat digunakan untuk sebanyak apapun produk yang dibangun[7].

Beberapa tim yang tergabung dalam *design system* yaitu designer, developers, dan produk manager. Memahami tujuan utama *design system* yaitu menerapkan solusi yang membantu tim mengukur produk dengan sukses. Dengan sistem yang jelas, desainer dan developers dapat lebih fokus pada permasalahan pengguna, dibanding harus menciptakan kembali elemen dan mencari kembali solusi[1].

Dalam membangun *design system* ada tiga bagian fondasi utama yang harus diperhatikan yaitu *foundation, component* dan *patterns* sebagai berikut [8] :

1. Foundations

*Foundations* merupakan level dasar dari semua produk yang ada termasuk *design system* itu sendiri. *Foundations* memiliki banyak sisi dan bisa dibilang lebih luas daripada tingkatan lainnya itulah sebabnya disbut lapisan dasar. *Foundations* biasanya mencakup sebuah *brand guidelines seperti typography, text styles, colour, icons, spacing* dan *layout.*

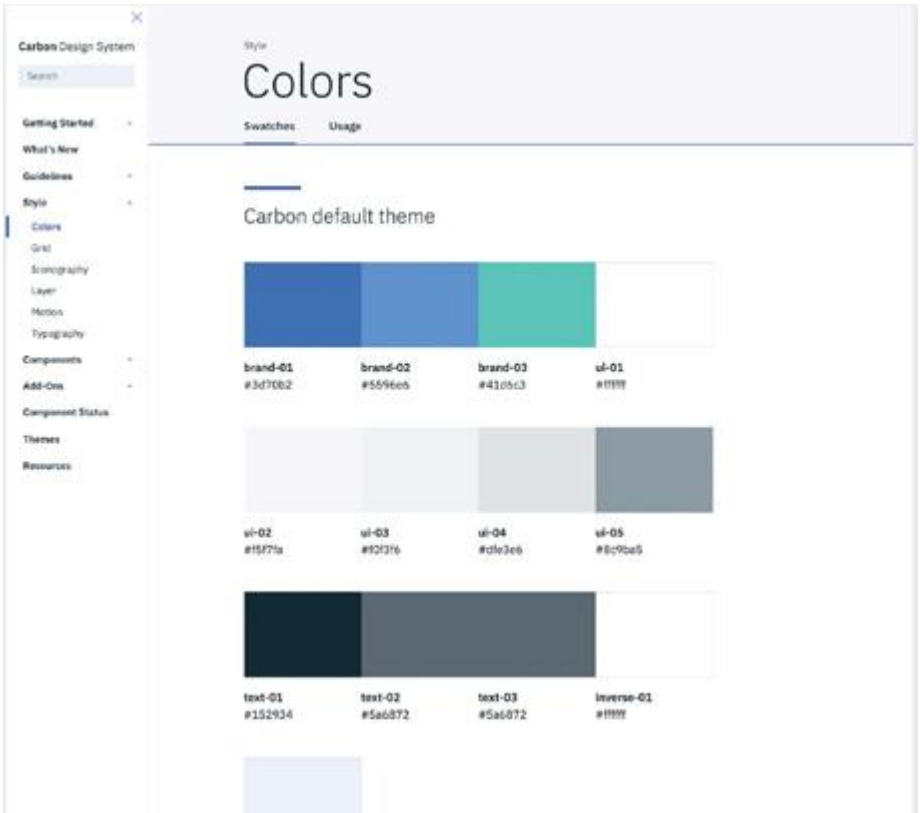
A. Color

Color (Warna) yang dipilih untuk *design system* lebih dari sekedar nilai tambah pada sebuah brand. Tapi sebuah warna dalam UI digunakan untuk menyampaikan :

- Feedback : Error dan status sukses
- Information : Bagan, grafik dan elemen wayfinding

- Hierarchy: Menampilkan urutan struktur melalui color dan typography

Pada umumnya sebuah warna dalam *design system* memiliki 1 sampai 3 warna utama yang mempresntasikan sebuah brand. Jika dari pilihan warna yang telah ditentukan tidak cocok untuk digunakan dalam warna link dan button, maka boleh menambahkan warna baru. Menggunakan warna yang sama pada link dan button akan lebih mudah untuk pengguna mengenali komponen.




**Gambar 2.2 Contoh Color Palette**

B. Typography

Dalam typography ada tiga hal penting yang harus diperhatikan. 3 hal tersebut adalah sebagai berikut :

1. Font dan Weights

Font yang dipilih harus berdampak besar terhadap brand dan user experience. Tetap mementingkan keterbacaan font yang akan digunakan dalam typhograpy. Menggunakan font yang sering digunakan seperti

Times New Romana, Helvetica atau Verdana bisa menjadi pilihan yang bagus karena tidak asing di mata pengguna.

Pada umumnya para designer lebih sering menggunakan dua tipe font yang satu digunakan untuk judul dan yang satu untuk bagian isi. Design system tidak membutuhkan banyak tipe font karena akan menimbulkan masalah ketika terlalu banyak font yang digunakan. Gambar 2.3 merupakan contoh font weight.


**Gambar 2.3 Font Weights**

2. Type Scale

Saat memilih ukuran untuk menyesuaikan dengan font yang digunakan, harus mepertimbangkan keterbacaannya dari font yang dipilih. Dalam kebanyakan kasus, ukuran font 16px bagus untuk diguanakan. 16px adalah ukuran font default di sebagian besar browser, dan memang begitu cukup mudah dibaca oleh kebanyakan orang.

Untuk menentukan ukuran font yang lebih besar atau yang lebih kecil bisa menggunakan modularscale. Seperti yang terlihat dalam Gambar 2.4


**Gambar 2.4 Modular Scale**

Ketika menentukan ukuran font harus memperhatikan bagaimana font bekerja dalam responsive karena agar tidak terlalu besar ketika dibuka di perangkat seluler dan tidak terlalu kecil saat di buka di ukuran desktop.

2. Leading

Leading atau line-heights bisa meningkatkan readability dan nilai estetika dari typography yang digunakan. Secara aturan umum ukuran leading yang bagus sekitar 1,4-1,5x. Ukuran 1,5 direkomendasikan oleh W3C Web Accessibility Initiative untuk digunakan dalam leading font.

C. Spacing dan Sizing

Sistem yang digunakan untuk *spacing* dan *sizing* akan terlihat bagus ketika memiliki ritme dan keseimbangan yang mana maksudnya menggunakan angka berdasarkan *pattern* dan *proportions* dan bagian penting dalam membuat *visual design* [9]. Menggunakan skala spacing juga bisa meningkatkan *maintability* dan membuat layout akan lebih pas dan selaras.

Skala 4 menjadi dasar yang sangat paling banyak digunakan dengan berbagai banyak alasan. iOS dan Android merekomendasikan metrik yang

dapat dibagi atau kelipatan 4. Standard ICO size formats, yang mana sering digunakan oleh kebanyakan sistem operasi, untuk ikon cenderung berbasis 4 (16, 24, 32, dst) sehingga bisa menskalakan dengan lebih mudah. Ukuran font *default* browser pada umumnya menggunakan ukuran 16px.

D. Imaging

Dalam penggunaan image ada 3 hal yang harus diperhatikan. 3 hal tersebut sebagai berikut :

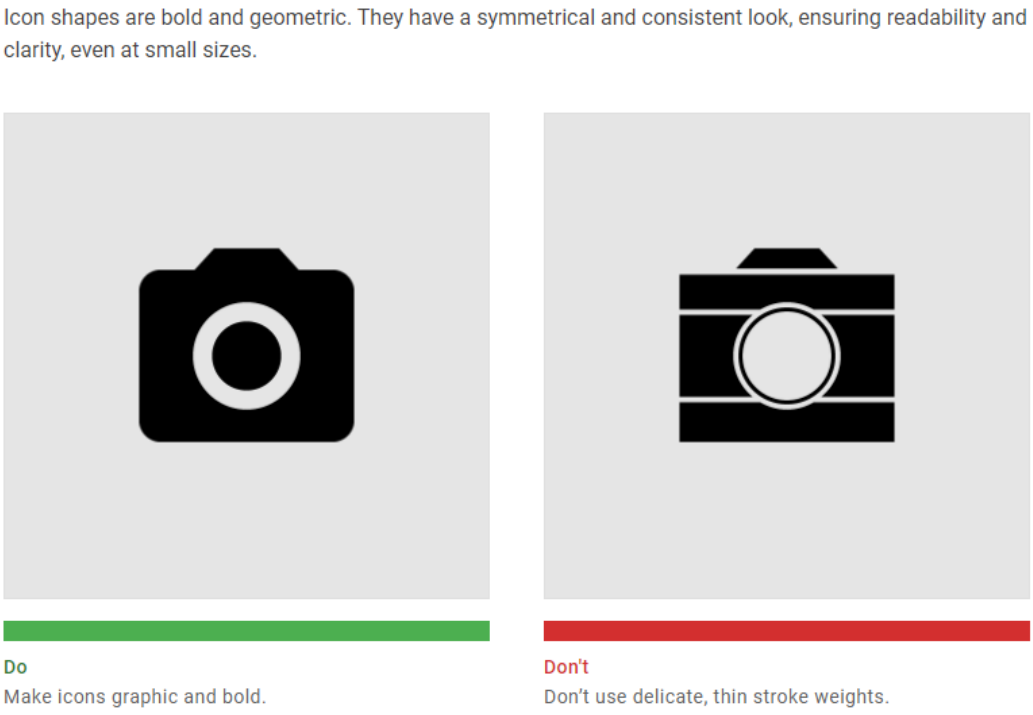
1. Format File

   Untuk ikons dan ilustrasi lebih baik menggunakan format SVG yang mana bagus untuk *scalability* dan *responsive.* Untuk menggunakan asset foto lebih baik menggunakan format seperti PNG dan JPG.

2. *Ikonography*

   Ketika membuat ikon terlebih dahulu harus memiliki aturan dan ada beberapa hal penting yang harus diperhatikan seperti ikon yang dibuat akan dalam bentuk *fill* atau *outline*, *line weights* yang diterapkan, jumlah warna yang akan dipakai, menentukan ukuran ikon.

   Kebutuhan ikon mungkin akan berbeda di setiap tipe ikon yang digunakan contohnya seperti ikon *utility* dan ikon *action* seperti ikon notifikasi atau ikon konfigurasi yang memiliki warna yang solid dengan satu warna, berbeda untuk ikon navigasi yang memiliki lebih banyak warna agar terlihat lebih kreatif. Dengan aturan yang jelas ikon akan tetap konsisten. Seperti yang di perlihatkan pada Gambar 2.5.




**Gambar 2.5 Ikon Material Design**

2. Components

Components merupakan bagian terkecil *building block* dari sebuah produk. Components merupakan sebuah tampilan antarmuka yang digunakan berulang kali di semua produk. Kebanyakan *component* berupa sebuah aksi atau digunakan untuk menyampaikan sesuatu contoh seperti *buttons, form inputs, selects, textareas, radio buttons,checkboxes, range sliders, toggles, avatars, tooltips*, dan lainnya.

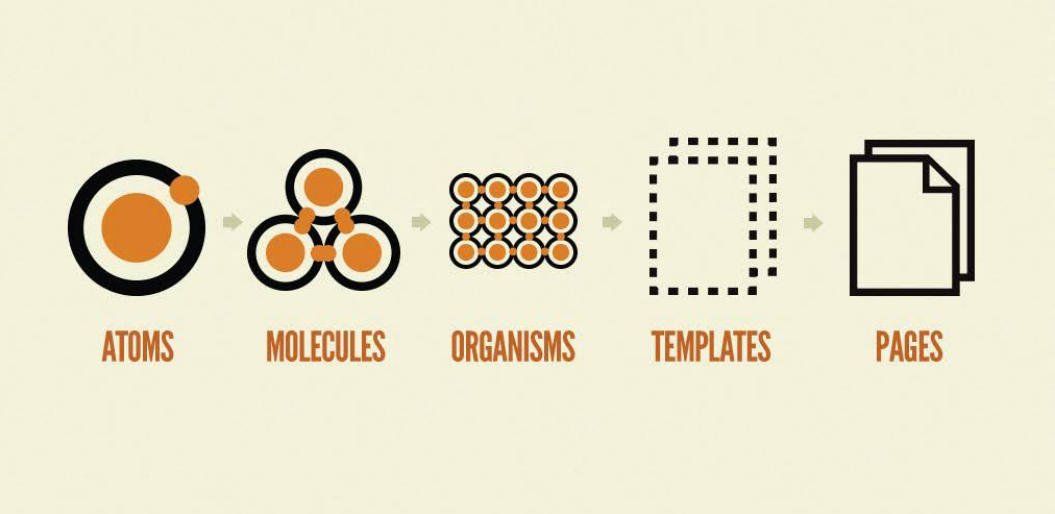
3. Patterns

Patterns merupakan bagian terbesar *building block* dari sebuah produk. Patterns lebih mengacu kepada elemen yang berulang pada semua produk.Sebuah *patterns* terdiri dari kumpulan *foundation* dan *components* sebagai contoh seperti card yang mungkin terdiri dari sebuah header, deskripsi, menggunakan beberapa text style dan sebuah button contoh lainnya seperti navigation, footer, modals, alerts, notifications, tables, feed cards, product cards, image galleries, carousels, feature or masthead areas, pagination, breadcrumbs, dan lainnya

### 2.2.2 Atomic Design

Atomic Design merupakan metodologi untuk membangun design system yang dikembangkan oleh Brad Frost. Brad Frost mengatakan atomic design hasil riset dari luar bidang industri desain, terinpirasi dari ketika dia berada di kelas kimia waktu sekolah. Mengambil dari tiga unsur utama dalam kimia yaitu atom, moleculs dan organisms. Di alam, unsur-unsur atom bergabung membentuk molekul. Molekul-molekul ini dapat bergabung dengan membentuk organisme yang relatif kompleks. Brad Frost juga mengatakan bahwa semua elemen yang ada di dunia bisa di pecah sampai ke bagian terkecil. Sama seperti dalam membangun design system yang berawal dari kumpulan *color, image, typography, spacing dan sizing, dan visual form* yang bisa dibuat menjadi kumpulan komponen yang kompleks [10].

Atomic design bukan sebuah proses linear, tetapi lebih ke arah mental model untuk membantu memikirkan *user interface* sebagai satu kesatuan yang utuh dan kumpulan komponen yang bisa digunakan dalam waktu yang bersamaan. Atomic design merupakan metodologi yang terdiri dari lima tahapan yang saling bekerja satu sama lain untuk membuat interface design system. Setiap tahapan mempunyai peranan penting di dalam hirarki interface design system. Ke lima tahapan tersebut seperti yang terlihat didalam Gambar 2.6

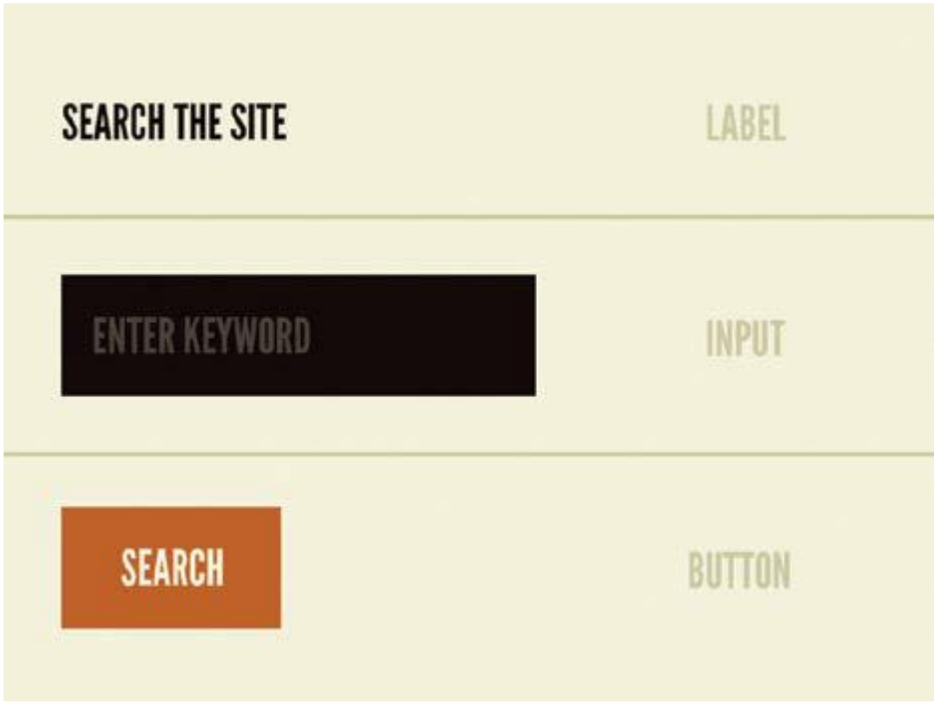



**Gambar 2.6 Tahapan Atomic Design**

1. Atoms

   Atoms adalah bagian terkecil yang sangat penting pada suatu materi yang menjadi dasar untuk membangun sebuah tampilan user interface. Elemen basic HTML termasuk juga ke dalam bagian dari atoms seperti form labels,

inputs, buttons, dan lainnya yang tidak mungkin dipecah lagi. Seperti yang terlihat pada Gambar 2.7

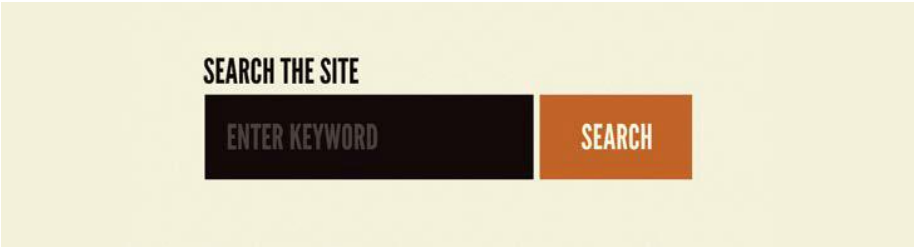



**Gambar 2.7 HTML Tag**

Setiap komponen dalam atoms memiliki ciri yang unik seperti pada bagian hero image, atau ukuran font pada heading utama. Dalam konteks pattern library atoms hanya terlihat sebagai style dasar yang sangat berguna untuk membantu dalam pengembangan dan perawatan design system.

2. Moleculs

Dalam kimia moleculs merupakan kumpulan komponen-komponen atom yang saling terikat satu sama lain lalu menciptakan komponen baru yang berbeda. Moleculs adalah kumpulan komponen UI yang relatif sederhana dan berfungsi sebagai satu kesatuan seperti yang terlihat pada Gambar 2.8




**Gambar 2.8 Form Search**

Setelah menggabungkan komponen atoms yang bersifat abstrak sekarang memiliki fungsi. Label atom menjadi definisi dari imput atom. Button klik atom menjadi button untuk submit. Komponen yang sederhana, portable, reusable dapat ditempatkan di fungsi pencarian mana saja. Membuat komponen yang sederhana dapat memudahkan UI designer dan developers untuk mematuhi aturan. Membebani satu *pattern* dengan terlalu rumit membuat perangkat lunak menjadi berat. Maka dari itu membuat komponen molecul UI yang sederhana dapat mempermudah melakukan testing, bisa digunakan kembali dan meningkatkan konsistensi semua *user interface.*

3. Organism

Organism adalah kumpulan komponen UI relatif kompleks terdiri dari kumpulan molecul, atom, dan bahkan dari organisms lainnya. Seperti organism header pada Gambar 2.9 dari kumpulan form search molecul, logo atom dan navigasi utama molecul.

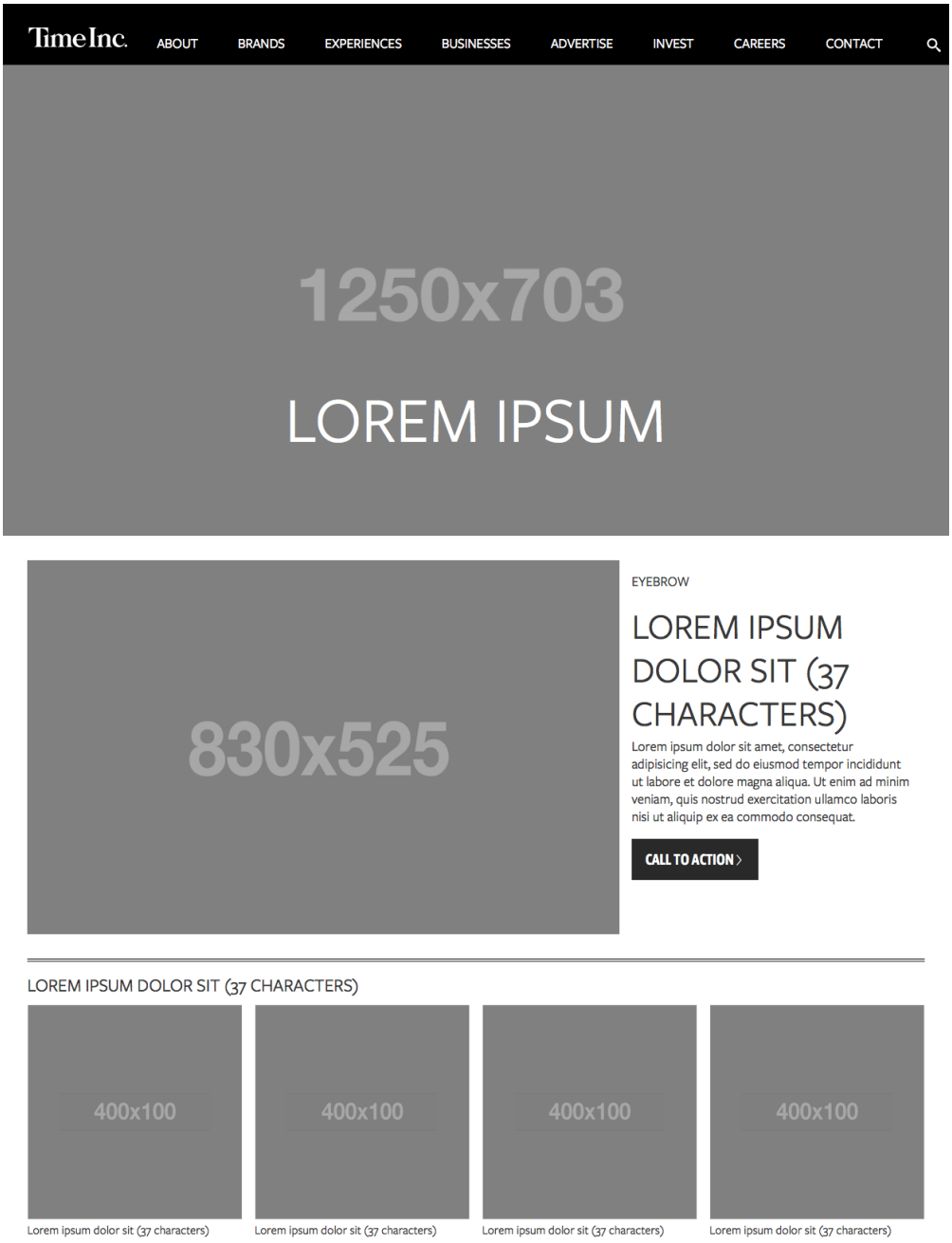

**Gambar 2.9 Organisms Header**

Organisms bisa terdiri dari molecul yang serupa atau berbeda. Sebuah header organisms mungkin terdiri dari elemen yang berbeda seperti logo gambar, daftar navigasi utama, dan form pencarian. Tipe organism ini sering terlihat di hampir setiap situs web. Organisms menampilkan komponen yang lebih kecil, sederhana dan berfungsi sebagai *pattern* yang berbeda tapi bisa digunakan berulang kali.

4. Templates

Templates adalah objek tingkatan halaman yang terdiri dari organisms dan menempatkan organism kedalam sebuah *layout*. Walaupun sudah dibuat menjadi sebuah halaman tetapi komponen-komponen atom, molecul dan organisms belum diberikan warna. Dalam template lebih berfokus kepada

penempatan setiap komponen dari atom, moleculs, dan organism agar menjadi sebuah tampilan halaman yang padu seperti yang terlihat pada Gambar 2.10.




**Gambar 2.10 Template**

5. Pages

Pages adalah hal yang lebih spesifik dari template dimana meperlihatkan seperti apa UI yang dilihat nanti oleh pengguna dengan

menggunakan konten nyata seperti images, color, icon dan lainnya. Selain pages akan menjadi tampilan terakhir yang dilihat oleh pengguna, pages juga sangat penting untuk menguji keefektifan dari *design system*. Jika hasil dari testing tidak memberikan hasil semestinya maka harus melakukan kembali modifikasi dari molecules, organisms, dan templates sampai sesuai dengan apa yang dibutuhkan. Contoh tampilan pages bisa dilihat pada Gambar 2.11.


**Gambar 2.11 Pages**

### *2.2.3 Design Audit*

Audit Design merupakan sebuah metode dalam melakukan analisis semua komponen desain yang digunakan dalam sebuah produk dimana dilakukan untuk mengumpulkan semua aset yang ada pada semua platform untuk memastikan sebuah brand memiliki konsistensi dalam produknya [11]. Konsistensi dalam desain sangat berpengaruh terhadap pengalaman kostumer. Begitu juga pada sebuah *user interface* (UI) ketika sebuah UI memiliki konsistensi akan mudah di prediksi yang artinya pengguna dapat memahami bagaimana menggunakan fungsi tertentu secara intuitif dan tanpa instruksi. Membuat sebuah produk mudah untuk digunakan merupakan sebuah pencapaian untuk membuat produk lebih diinginkan. Sebaliknya ketika sebuah UI tidak memiliki konsistensi dapat membuat pengguna susah untuk memahami yang bisa menimbulkan rasa frustasi dan menimbulkan sebuah *user experience* yang buruk [2].

### *2.2.4 Interface Inventory*

Interface Inventory merupakan sebuah cara yang sama dengan *design audit* yaitu mengumpulkan semua komponen desain yang digunakan pada sebuah produk dengan tujuan mengumpulkan semua aset tapi yang membedakan dalam interface inventory yaitu mengkategorikan setiap komponen yang mebuat website, aplikasi atau lainnya. Dalam melakukan intervace inventory ada beberapa tahapan yang dilakukan yaitu :

1. Pertama buka projek, aplikasi, website atau lainnya yang berhebungan dengan konten
2. Siapkan sebuah template kosong digunakan untuk menaruh dan mengkategorikan semua komponen antarmuka. Alat yang digunakan bisa berupa power point atau apapun.
3. Mulai melakukan *screenshooting* semua bagian komponen tampilan antarmuka seperti *headings, text fields, radio buttons, carousels, accordions, tabs, images, icons, video players, graphs,* dan lainnya. Perhatikan bahwa komponen yang diambil tidak sama harus memiliki prilaku yang berbeda dar setiap komponen yang diambil.

4. Kategorikan hasil dari *screenshoot* untuk dapat melihat semua prilaku dari komponen yang ada.

### *2.2.5 Perceptual Pattern*

*Perceptual Pattern* merupakan sebuah level abstrak dari sebuah kumpulan *pattern* yang digunakan sebagai aturan dasar atribut *style* yang digunakan dalam sebuah produk atau sebuah sistem. *Perceptual pattern* meliputi beberapa aturan visual dari sebuah elemen dasar seperti *color, typography, spacing & layout, icon, ilustration* dan *style guidelines* yang merupakan sebuah aturan dasar untuk desainer dalam membuat *user interface* yang disediakan dalam bentuk dokumentasi[6].

### *2.2.6 Functional Pattern*

*Functional Pattern* adalah sekumpulan elemen komponen user interface yang memiliki pola dari tampilan antarmuka. Tujuannya adalah untuk mendorong kebiasaan atau perilaku pengguna pada saat kondisi tertentu [6]. Functional pattern harus di definisikan agar behavior dan mental model yang dimiliki pengguna terhadap tampilan antarmuka dapat ditemukan.

### 2.2.7 Material Design

Material Design adalah design system yang diciptakan oleh Google untuk membantu tim membangun produk digital berkualitas tinggi berdasarkan pengalaman, material design tersedia untuk Android, iOS, Flutter, dan web. Material design adalah sistem guidelines yang bisa beradaptasi, kumpulan komponen, dan *tools* yang bagus untuk desain user interface. Material design memudahkan designer dan developer berkolaborasi dan membantu tim dengan cepat untuk membuat tampilan UI yang indah. Dalam material design banyak sekali panduan yang berguna untuk membuat komponen-komponen yang dibutuhkan dalam membangun design system dengan menggunakan prinsip material design itu sendiri [12].

### 2.2.8 Usability Metrics

Usability Metrics berdasarkan ISO *Organization for Standardization* (9241-11) Sejauh mana sebuah produk dapat digunakan oleh pengguna tertentu untuk mencapai tujuan tertentu dengan efektivitas, efisiensi, dan kepuasan dalam konteks penggunaan yang ditentukan. Pengguna tertentu: bukan sekedar siapa penggunanya tetapi untuk siapa produk tersebut di desain, Tujuan tertentu: pengguna tertentu harus berbagi tujuan untuk product, artinya produk tersebut mewakili tujuan mereka, Konteks penggunaan yang ditentukan: Produk yang dibuat harus dirancang di lingkungan yang mana pengguna akan menggunakanya[13]. Tiga hal penting lainnya dari pengertian ISO (9241-11) adalah *effectiveness*, *efficiency*, dan *satisfaction* yang merupakan model standar ISO dari tigal hal tersebut mempunyai makna yang berbeda berikut penjelasanya [14] :

- *Effectiveness* : Berapa banyak pengguna menyelesaikan tugasnya
- *Effieciency* : Berapa lama pengguna menyelesaikan tugasnya
- *Satisfaction* : Seberapa nyaman pengguna menggunakan produk tersebut [15]

Dalam pengujian *design system* akan melakukan penekanan aspek terhadap *usability testing* secara *effieciency* yang mana digunakan untuk menghitung seberapa lama pengguna dapat menyelesaikan suatu tugas. Dalam mengukur tingkat efisiensi dari *design system* digunakan perhitungan secara *overall relative efficiency* dengan menggunakan rasio waktu yang dibutuhkan pengguna yang berhasil menyelesaikan tugasnya. Perhitungan tersebut dapat digambarkan dalam persamaan dibawah ini [14][16][17] :

$$ORE = \frac{\sum_{j=1}^{R}\sum_{i=1}^{N} n_{ij}t_{ij}}{\sum_{j=1}^{R}\sum_{i=1}^{N} t_{ij}} \times 100$$

Keterangan :

ORE = *Overall Relative Efficiency*

N = Jumlah total tugas

R = Jumlah pengguna

nij = Hasil tugas i oleh pengguna j; jika pengguna berhasil menyelesaikan tugas, maka nij = 1, jika tidak, maka nij = 0

tij = Waktu yang dihabiskan oleh pengguna j untuk menyelesaikan tugas i. Jika tugas tidak berhasil diselesaikan, maka waktu diukur hingga saat pengguna berhenti.