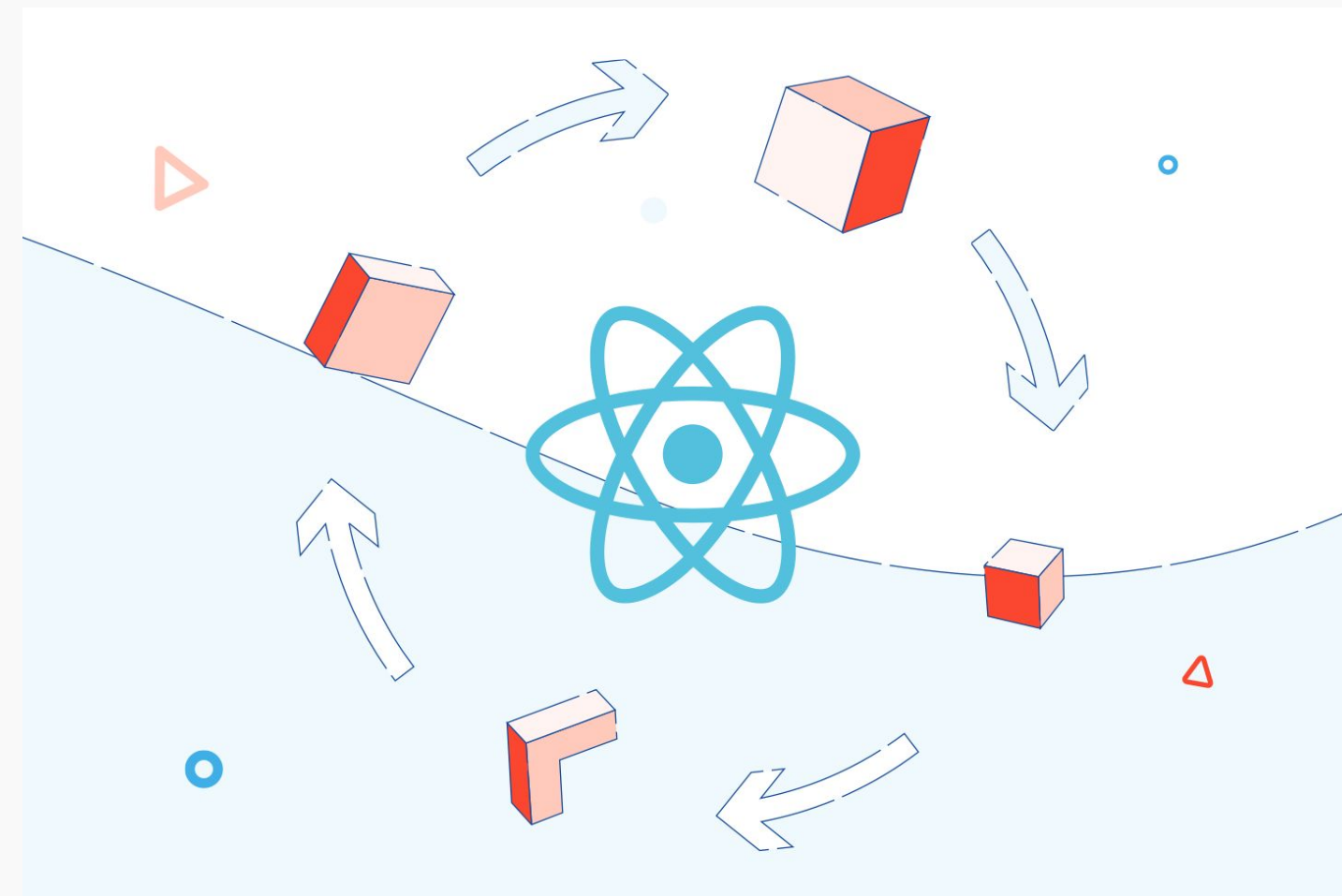


Google Developer Group
Universitas Sriwijaya

Learning 4

Components & Props

Build reusable UI with data

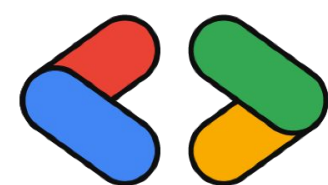
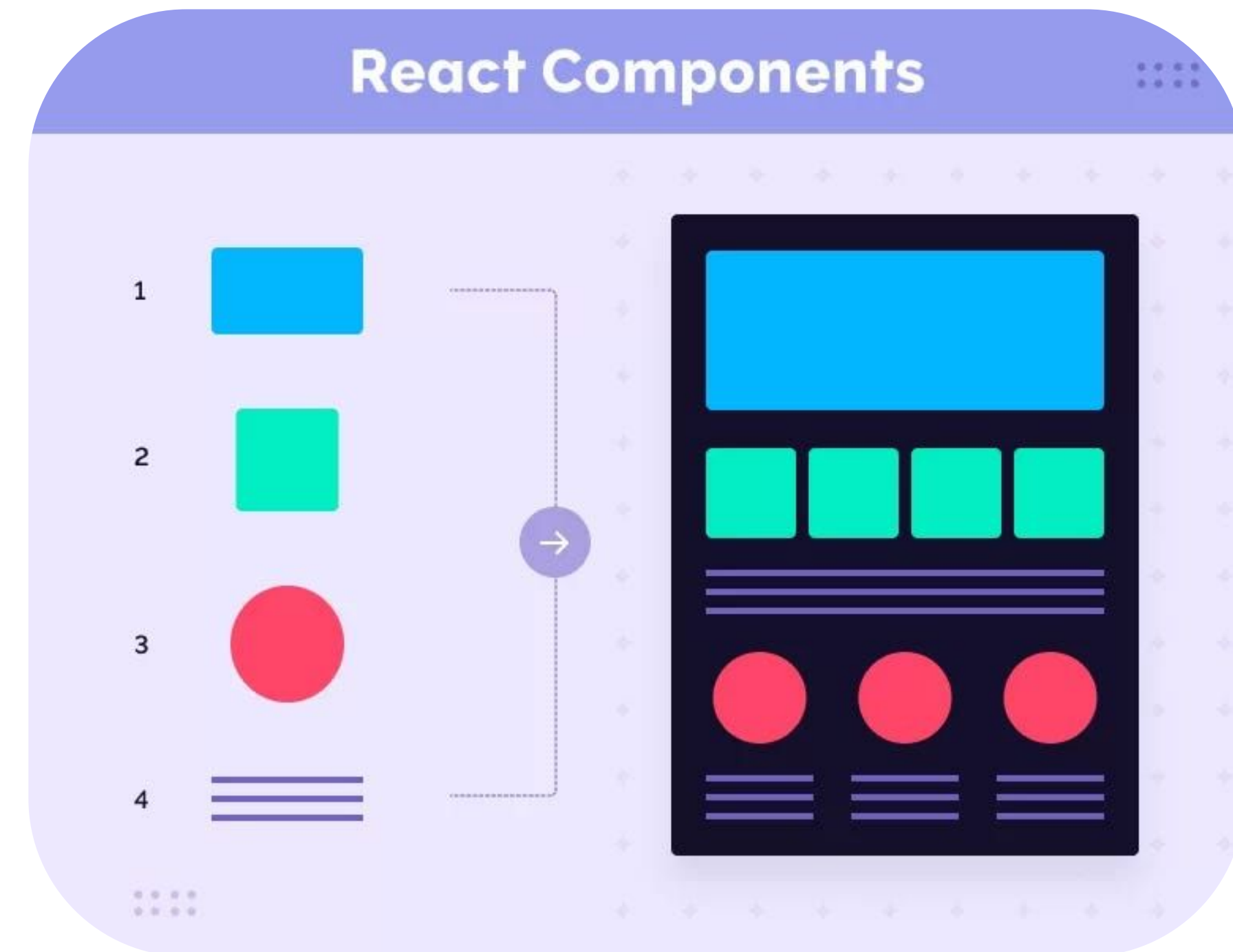


```
const org = interbyOrg ? study.lead_organization === interbyOrg : C  
const status = filterByStatus ? study.status === filterByStatus : true  
const matchStatus = {  
  matchStatus  
}
```

```
function filterStudies({ studies, filterByOrg =  
  filterByOrg, filterByStatus = filterByStatus }) {  
  return studies.filter(study => {  
    const org = interbyOrg ? study.lead_organization === interbyOrg : C  
    const status = filterByStatus ? study.status === filterByStatus : true  
    const matchStatus = {  
      matchStatus  
    }  
    return org & status & matchStatus  
  })  
}
```

React Components

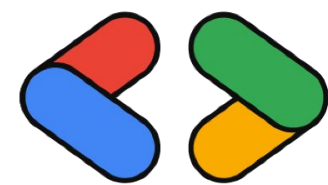
- Component = bagian UI
- Dibuat sebagai function
- Bisa digunakan berulang



What is Props?

- Props = data yang dikirim ke component
- Bersifat read-only
- Digunakan untuk membuat UI dinamis

```
<Card title="Product A" />
```



Props as Function Parameter

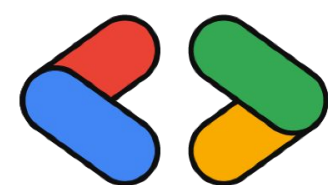
- Component menerima props sebagai parameter
- Props digunakan di dalam JSX
- Bisa di-destructure

Without Destructuring

```
function Card(props) {  
  return <h1>{props.title}</h1>  
}
```

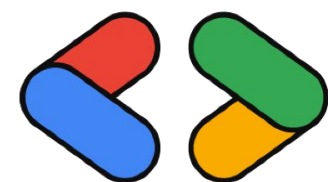
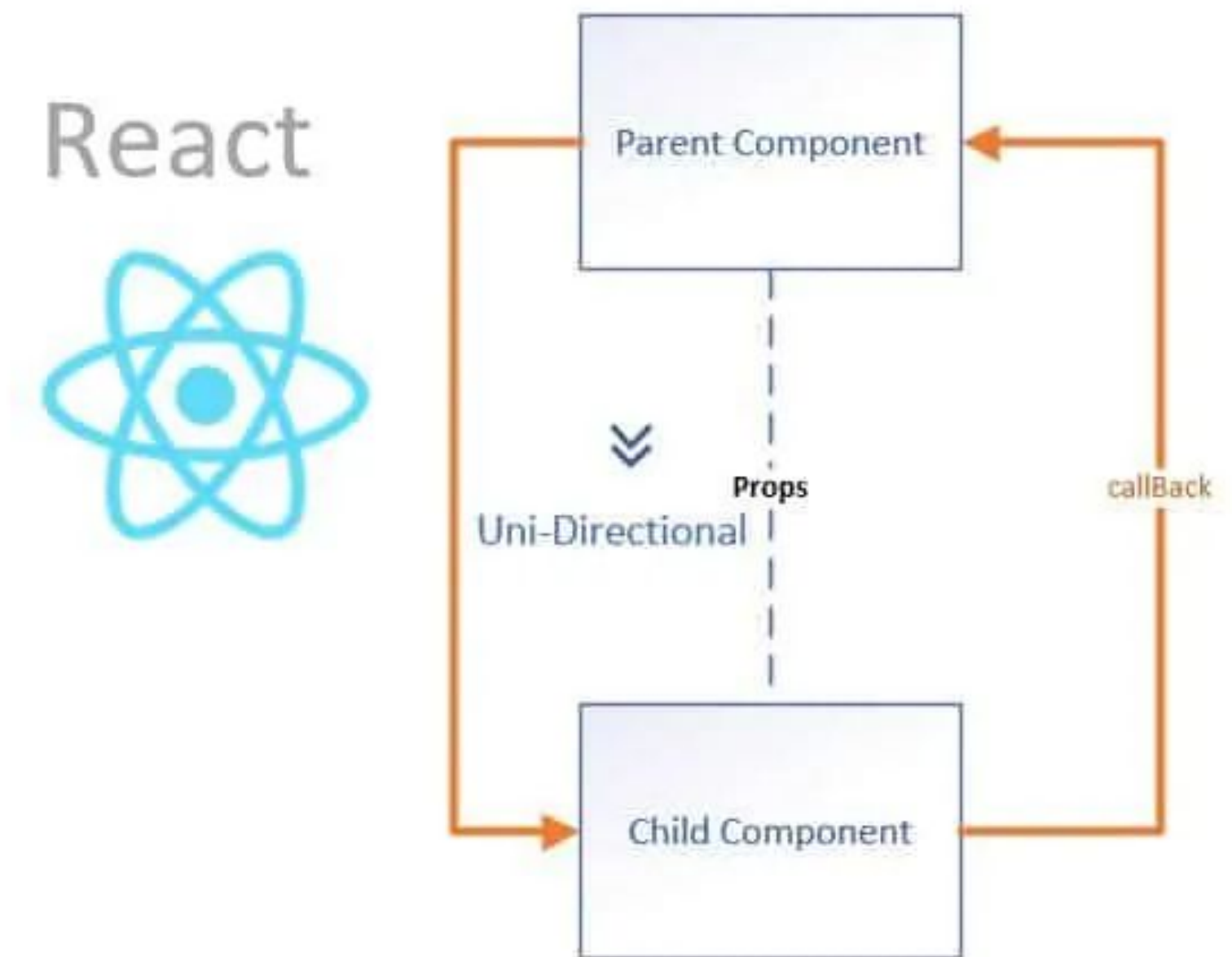
With Destructuring

```
function Card({ title }) {  
  return <h1>{title}</h1>  
}
```



Parent → Child Communication

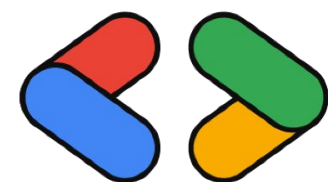
- Data dikirim dari parent ke child
- Menggunakan props
- Alur data satu arah (one-way)



Reusable Component

- *Satu component*
- *Digunakan berkali-kali*
- Data berbeda, tampilan sama

```
<Card title="Product A" />  
<Card title="Product B" />
```



Component Children

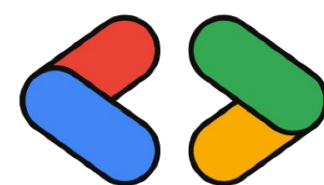
- Children adalah props khusus
- Berisi konten di dalam component
- Membuat component lebih fleksibel

Component

```
function Card({ children }) {  
  return <div>{children}</div>  
}
```

Component Usage

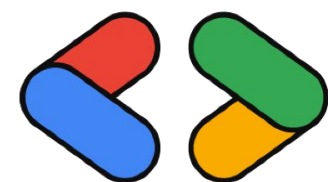
```
<Card>  
  <p>Hello World</p>  
</Card>
```



Rendering List with map()

- Data biasanya berbentuk array
- React menggunakan map() untuk looping
- Setiap item menghasilkan component

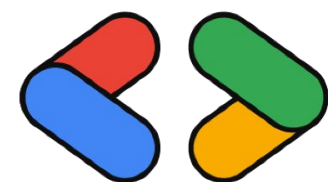
```
products.map(product => (  
  <Card title={product.name} />  
))
```



Key in List Rendering

- Key wajib saat rendering list
- Membantu React mengenali element
- Gunakan id yang unik

```
<Card key={product.id} />
```



Conditional Rendering

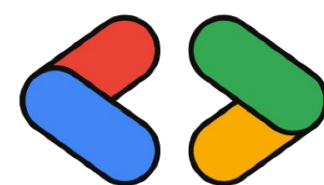
- Menampilkan UI berdasarkan kondisi
- Menggunakan JavaScript expression
- Membuat UI lebih dinamis

And Operator

```
{isLoggedIn && <Profile />}
```

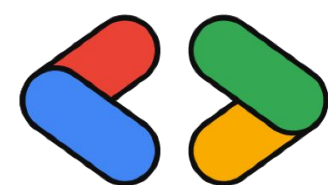
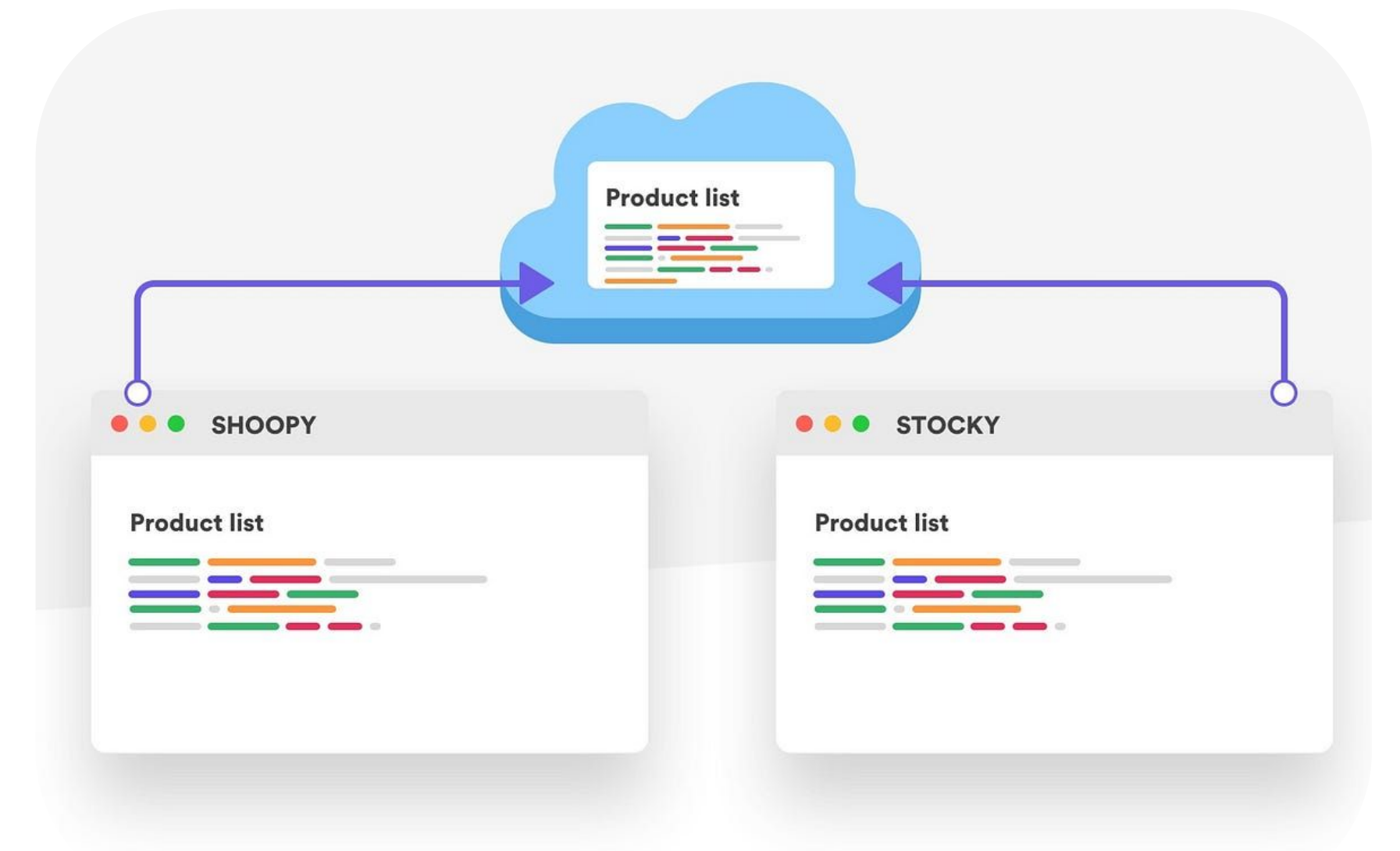
Ternary

```
{status ? <Success /> : <Error />}
```



Good Component Structure

- Satu component satu tanggung jawab
- Pisahkan UI besar menjadi komponen kecil
- Pisahkan component sesuai section/feature
- Nama component jelas & konsisten

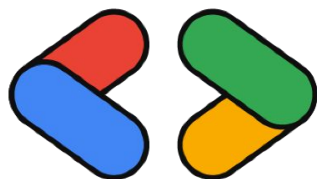


Let's Code!!!

```
const products = [
  { id: 1, name: "Product A", price: "100k" },
  { id: 2, name: "Product B", price: "200k" },
];

function Card({ name, price, children }) {
  return (
    <div className="border p-4 rounded">
      <h2>{name}</h2>
      <p>{price}</p>
      {children}
    </div>
  );
}

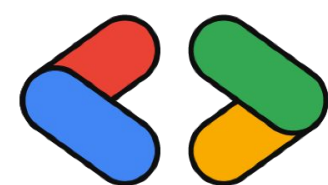
function App() {
  return (
    <div className="space-y-4">
      {products.map((item) => (
        <Card key={item.id} name={item.name} price={item.price}>
          <button>Buy</button>
        </Card>
      ))}
    </div>
  );
}
```



Task! (FE Member Only*)

1. Pahami ulang materi React Components dan Props
2. Pelajari penggunaan children pada component
3. Pelajari rendering list menggunakan map()
4. Eksplor conditional rendering (&&, ternary/?)
5. Review JavaScript ES6 yang sering digunakan di React
6. Praktik!!!

Selamat Mencoba ~



Google Developer Group
Universitas Sriwijaya



Thank You !