

## **BAB 3**

### **PELAKSANAAN KERJA MAGANG**

#### **3.1 Kedudukan dan Koordinasi**

Selama menjalani program magang di PT. Surya Digital Indonesia, peserta ditempatkan pada *IT Department* dan bekerja langsung di bawah arahan *Supervisor IT*. Dalam posisi tersebut, peserta menjadi bagian dari tim pengembang yang bertugas mengerjakan proyek pembuatan dan pengembangan *website* untuk salah satu klien perusahaan, yaitu sebuah toko besi.

Dalam pelaksanaan tugas sehari-hari, peserta magang berperan sebagai pendukung teknis, seperti membantu proses implementasi sistem, memperbaiki bagian kode yang membutuhkan perbaikan, serta melakukan pengujian terhadap fitur-fitur yang sudah dibuat. Seluruh kegiatan tersebut dilakukan dengan bimbingan dan koordinasi dari *Supervisor IT* sebagai penanggung jawab utama proyek.

Hubungan koordinasi antara peserta magang, *developer*, dan supervisor berlangsung secara terstruktur. Setiap keputusan, baik yang bersifat teknis maupun strategis, tetap mengikuti arahan *Supervisor IT*. Komunikasi kerja dilakukan secara rutin melalui rapat internal serta penyampaian laporan perkembangan proyek yang diberikan secara berkala.

##### **3.1.1 Alur Kerja dan Koordinasi Magang**

Kegiatan magang di Divisi IT PT. Surya Digital Indonesia berjalan dengan alur kerja yang rapi dan berfokus pada pencapaian hasil. Setiap proses koordinasi dalam tim dilakukan secara teratur agar setiap tahap pengembangan proyek dapat terlaksana sesuai jadwal dan standar yang telah ditentukan. Adapun alur kerja dan koordinasi selama magang dapat dijabarkan sebagai berikut:

###### **1. Tahap Perencanaan Proyek**

Pada tahap awal, *Supervisor IT* mengadakan *meeting* internal untuk memaparkan tujuan, ruang lingkup pekerjaan, serta target penyelesaian proyek *website* toko besi. Pada sesi tersebut dilakukan pembagian tugas kepada semua anggota tim, termasuk penentuan peran peserta magang berdasarkan kemampuan dan bidang yang dikuasai.

## **2. Tahap Perancangan dan Desain**

Setelah tahap perencanaan selesai, tim mulai menyusun rancangan struktur dan tampilan *website*, meliputi antarmuka pengguna (*user interface*) dan alur navigasi. Peserta magang bekerja bersama developer dalam menerapkan desain awal dan membangun struktur halaman menggunakan teknologi yang sudah ditetapkan, seperti *HTML*, *CSS*, dan *JavaScript*. *Supervisor IT* memberikan arahan teknis serta meninjau hasil desain sebelum proyek masuk ke tahap pengembangan.

## **3. Tahap Implementasi dan Pengembangan**

Pada tahap ini, peserta magang terlibat langsung dalam pembuatan fitur-fitur *website* berdasarkan rancangan yang telah disepakati. Setiap perkembangan pekerjaan dilaporkan secara berkala kepada developer dan *Supervisor IT* untuk mendapatkan evaluasi dan masukan. Koordinasi dilakukan melalui pertemuan rutin atau laporan mingguan guna memastikan seluruh anggota tim memahami progres proyek dengan jelas.

## **4. Tahap Pengujian dan Revisi**

Setelah proses pengembangan selesai, dilakukan pengujian (*testing*) untuk memastikan seluruh fitur berfungsi dengan benar dan tidak terdapat kesalahan. Peserta magang turut membantu proses *debugging* serta mencatat hasil pengujian. Jika ditemukan masalah teknis, *Supervisor IT* memberikan arahan untuk melakukan perbaikan sampai sistem berada pada kondisi stabil.

## **5. Tahap Finalisasi dan Dokumentasi**

Pada tahap akhir, tim melakukan finalisasi terhadap proyek *website* toko besi, termasuk penyusunan laporan hasil pekerjaan serta dokumentasi teknis. Peserta magang berkontribusi dalam pembuatan dokumentasi kode, panduan penggunaan, dan catatan revisi sistem. Seluruh dokumen kemudian dirangkum oleh developer dan diperiksa kembali oleh *Supervisor IT* sebelum diserahkan kepada pihak klien.

Alur kerja dan pola koordinasi yang tersusun dengan baik ini mencerminkan profesionalisme Divisi IT PT. Surya Digital Indonesia dalam menjalankan tiap proyek. Dengan bimbingan langsung dari *Supervisor IT*, kegiatan magang dapat berlangsung efektif dan memberikan pengalaman nyata terkait proses pengembangan produk digital untuk kebutuhan klien.

### **3.2 Tugas yang Dilakukan**

Selama menjalani kegiatan magang di PT Digital Go sebagai Frontend Developer, proyek yang dikerjakan dapat diselesaikan secara rutin setiap hari. Tanggung jawab yang diberikan dalam proses pengembangan website *e-commerce* Material Inovasi Industri meliputi:

1. Merancang tampilan antarmuka (*frontend*) dari website *e-commerce*.
2. Mengimplementasikan desain *frontend* ke dalam bentuk aplikasi web yang fungsional.

Tugas utama yang dilakukan selama kegiatan magang di PT Digital Go adalah pengerjaan *frontend* website *e-commerce* Material Inovasi Industri.

### **3.3 Uraian Pelaksanaan Magang**

Berikut adalah uraian pelaksanaan magang di PT Digital Go yang di lakukan selama 4 bulan kerja. Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.1.

Tabel 3.1. Rekapitulasi tugas mingguan selama magang

Minggu	Kegiatan
1	Mengembangkan bagian frontend Material Inovasi Industri serta melakukan revisi awal pada homepage dan beberapa proyek lain.
2	Fokus pada revisi homepage, cart page, serta penyempurnaan tampilan halaman produk di bagian frontend.
3	Mengerjakan revisi dashboard admin, memperbaiki dan mengembangkan halaman order confirmation (termasuk versi mobile), serta menyempurnakan sistem bidding price.
4	Mengembangkan checkout page, membangun halaman quotation-invoice-order pada admin panel, serta melakukan revisi pada bagian invoice dashboard.
5	Melanjutkan pengembangan halaman quotation, invoice, dan order pada admin panel serta melakukan revisi tampilan checkout dan quotation page di frontend.

Tabel 3.1. Rekapitulasi tugas mingguan selama magang (Lanjutan)

Minggu	Kegiatan
6	Mengembangkan halaman daftar produk dan generator invoice PDF, serta melakukan revisi lanjutan pada quotation dan invoice page di admin dashboard.
7	Melakukan revisi checkout pesanan, memperbaiki halaman driver pada admin dashboard, dan mulai mengembangkan fitur halaman tambah produk.
8	Mengembangkan halaman tambah produk dan customer page pada admin dashboard, serta merevisi panel quotation review dan generator invoice PDF.
9	Menyelesaikan revisi invoice PDF generator dan mulai mengerjakan fitur photobooth, termasuk revisi photosession page dan pembuatan fitur filter foto.
10	Melakukan revisi navbar e-commerce serta menyempurnakan tampilan pemilihan foto dan frame pada software photobooth.
11	Fokus pada perbaikan masalah printing foto, revisi tampilan photo result, serta penyempurnaan fitur payment dan photo preview pada software photobooth.
12	Melakukan perbaikan lanjutan pada website photobooth serta revisi halaman transaction dan admin panel pada e-commerce Material Inovasi Industri.
13	Mengoptimalkan admin panel photobooth, termasuk revisi pada payment section, page setting, serta peningkatan tampilan keseluruhan aplikasi.
14	Menyelesaikan revisi halaman setting photobooth dan melakukan pembaruan pada quotation section di admin panel e-commerce.

### 3.3.1 Perancangan

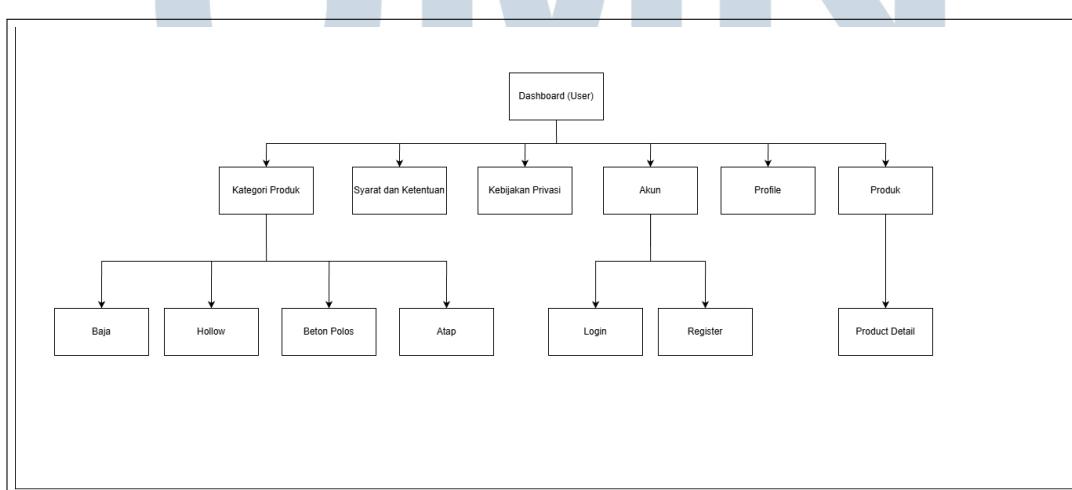
Pada sub bab ini, akan dijelaskan perancangan *website e-commerce* Material Inovasi Industri yang dikerjakan. Sub bab ini akan menjelaskan *sitemap* dan juga *flowchart*. *Website e-commerce* PT Material Inovasi Industri yang dikembangkan dalam kegiatan kerja praktik ini merupakan pengembangan lanjutan dari sistem yang telah ada sebelumnya. Oleh karena itu, proses perancangan difokuskan

pada penyesuaian dan penyempurnaan fitur yang telah tersedia, khususnya pada sisi antarmuka pengguna dan alur fungsional tertentu. Pengembangan lanjutan ini mencakup salah satunya penyesuaian alur pengecekan status *invoice*, peningkatan keterbacaan informasi pesanan, serta penyesuaian komponen antarmuka agar sesuai dengan kebutuhan pengguna.

## A Sitemap

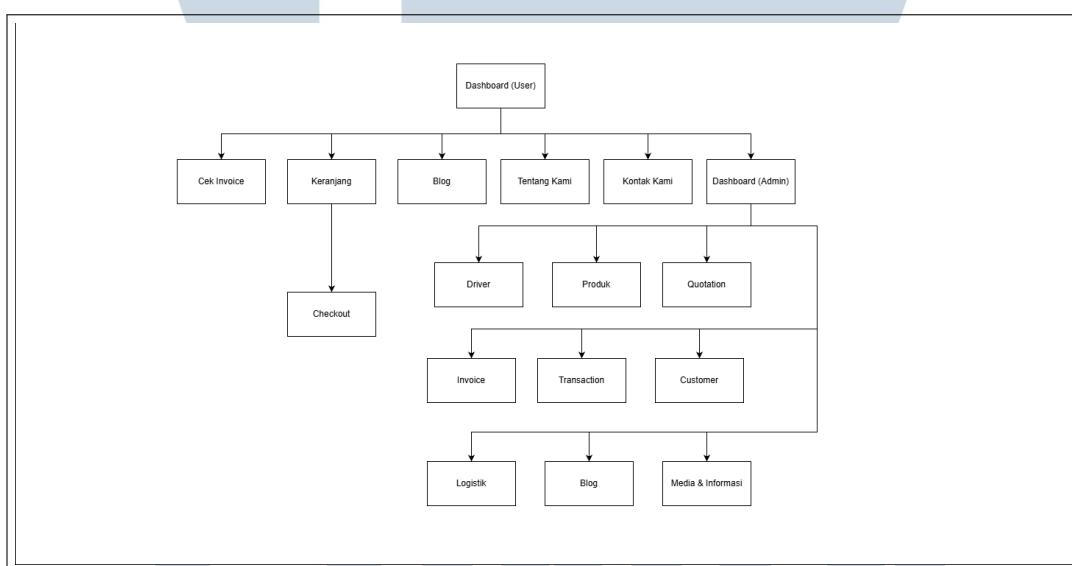
Berikut adalah *sitemap* dari *website e-commerce* material inovasi industri yang telah dibuat.

Sitemap pada gambar 3.1 menggambarkan struktur navigasi utama pada sisi pengguna (*user*) dalam *website* yang dikembangkan. Halaman utama dimulai dari *Dashboard (User)* sebagai pusat navigasi yang mengarahkan pengguna ke beberapa menu utama, yaitu Kategori Produk, Syarat dan Ketentuan, Kebijakan Privasi, Akun, *Profile*, dan Produk. Menu Kategori Produk berfungsi untuk mengelompokkan produk berdasarkan jenisnya, yang terdiri dari Baja, *Hollow*, Beton Polos, dan Atap, sehingga memudahkan pengguna dalam menemukan produk sesuai kebutuhan. Menu Akun menyediakan fitur autentikasi pengguna yang mencakup halaman *Login* dan *Register*. Menu *Profile* digunakan untuk menampilkan serta mengelola informasi data diri pengguna. Sementara itu, menu Produk menampilkan daftar produk yang tersedia dan menyediakan akses ke halaman *Product Detail* untuk melihat informasi produk secara lebih rinci. Selain itu, halaman Syarat dan Ketentuan serta Kebijakan Privasi disediakan untuk memberikan informasi terkait aturan penggunaan dan perlindungan data pengguna.



Gambar 3.1. *Sitemap e-commerce* material inovasi industri (bagian 1)

Sitemap pada gambar 3.2 menggambarkan fitur lanjutan pada website yang mencakup aktivitas pengguna dan pengelolaan sistem oleh admin. Dari *Dashboard (User)*, pengguna dapat mengakses menu Cek Invoice untuk melihat status transaksi, Keranjang yang terhubung ke halaman Checkout untuk proses pemesanan, serta halaman informatif seperti *Blog*, Tentang Kami, dan Kontak Kami. Selain itu, sistem menyediakan akses ke *Dashboard (Admin)* yang berfungsi sebagai pusat pengelolaan data. Pada dashboard admin terdapat beberapa modul utama, yaitu *Driver*, *Produk*, dan *Quotation*, yang masing-masing terhubung dengan pengelolaan *Invoice*, *Transaction*, dan *Customer*. Admin juga memiliki akses ke fitur pendukung lainnya seperti Logistik, *Blog*, serta Media dan Informasi yang digunakan untuk mengatur proses distribusi, konten, dan penyampaian informasi kepada pengguna.



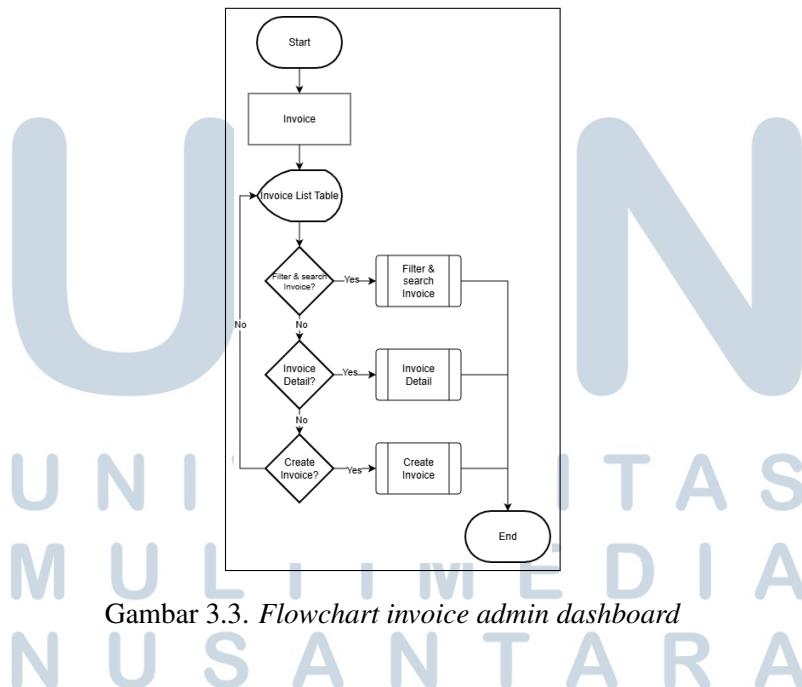
Gambar 3.2. Sitemap e-commerce material inovasi industri (bagian 2)

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

## B Flowchart

Berikut ini adalah *flowchart* dari beberapa *page* yang saya kerjakan selama magang di PT Digital Go.

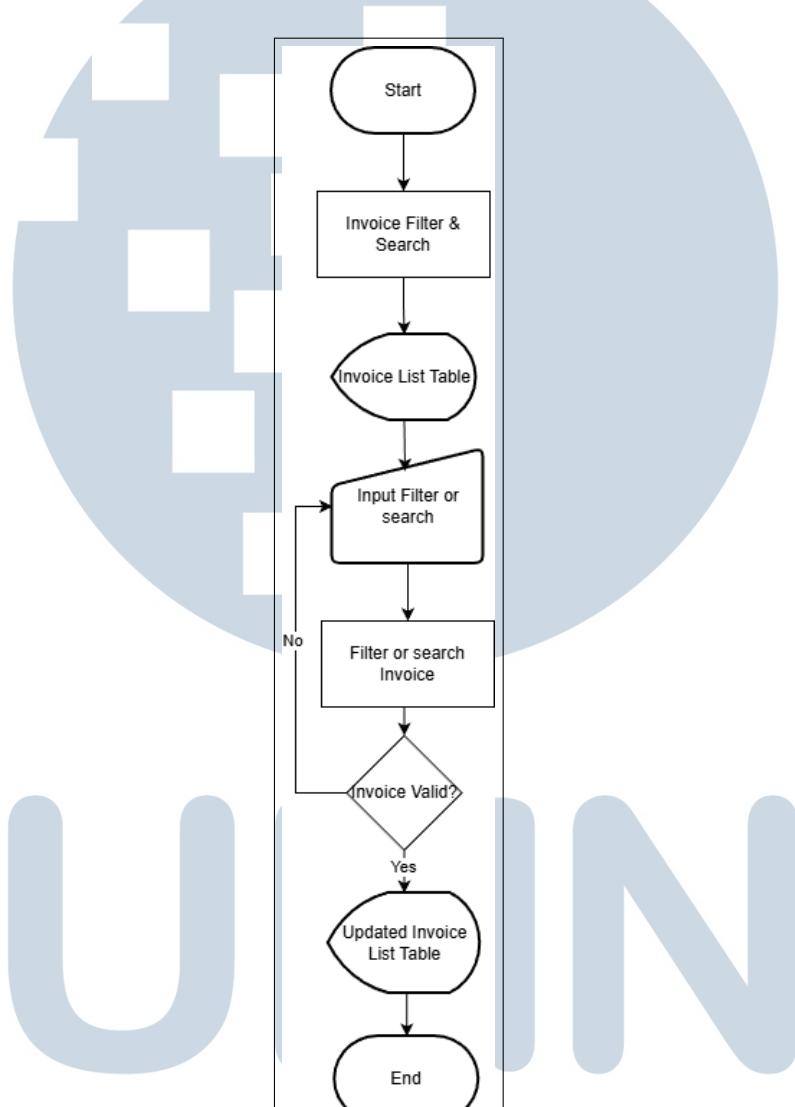
Gambar 3.3 merupakan *flowchart* yang menggambarkan alur *dashboard page Invoice* pada *Admin Dashboard*. Proses dimulai dari halaman *Invoice*, kemudian sistem menampilkan *Invoice List Table* sebagai tampilan utama daftar *invoice*. Pada halaman ini, pengguna dapat melakukan *filter* dan *search invoice*, melihat detail *invoice*, serta membuat *invoice* baru. Setiap aksi yang dipilih akan menampilkan halaman atau komponen terkait, sedangkan apabila tidak terdapat aksi yang dipilih, sistem akan tetap berada pada tampilan *Invoice List Table*. Alur ini merepresentasikan implementasi fitur pengelolaan *invoice* yang dikembangkan selama kerja praktik, yang bertujuan untuk mempermudah admin dalam memantau dan mengelola data transaksi secara terstruktur dan efisien. Data *invoice* yang ditampilkan pada halaman ini bersumber dari tabel *database invoices* dan *orders* sebagai penyimpan utama informasi transaksi dan status pembayaran, serta didukung oleh tabel *order\_items* untuk menampilkan detail item pesanan dan tabel *customers* untuk menampilkan informasi pelanggan terkait *invoice* tersebut.



Gambar 3.3. *Flowchart invoice admin dashboard*

Gambar 3.4 merupakan *flowchart* yang menggambarkan alur pada fitur *filter* dan *search invoice* di *Admin Dashboard*. Proses dimulai dari *Start*, kemudian sistem menampilkan halaman *Invoice* yang berisi *Invoice List Table* sebagai daftar utama *invoice*. Selanjutnya, pengguna melakukan input *filter* atau *search* melalui

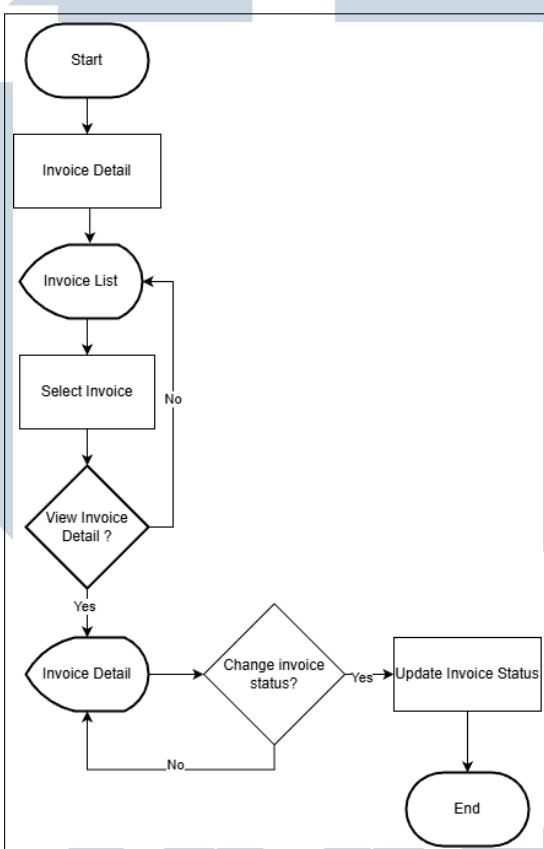
komponen yang tersedia. Sistem kemudian memproses penyaringan atau pencarian *invoice* sesuai dengan kriteria yang dimasukkan dan menampilkan hasilnya pada *Updated Invoice List Table*. Setelah daftar *invoice* diperbarui, alur proses berakhir pada *End*.



Gambar 3.4. Flowchart invoice filter admin dashboard

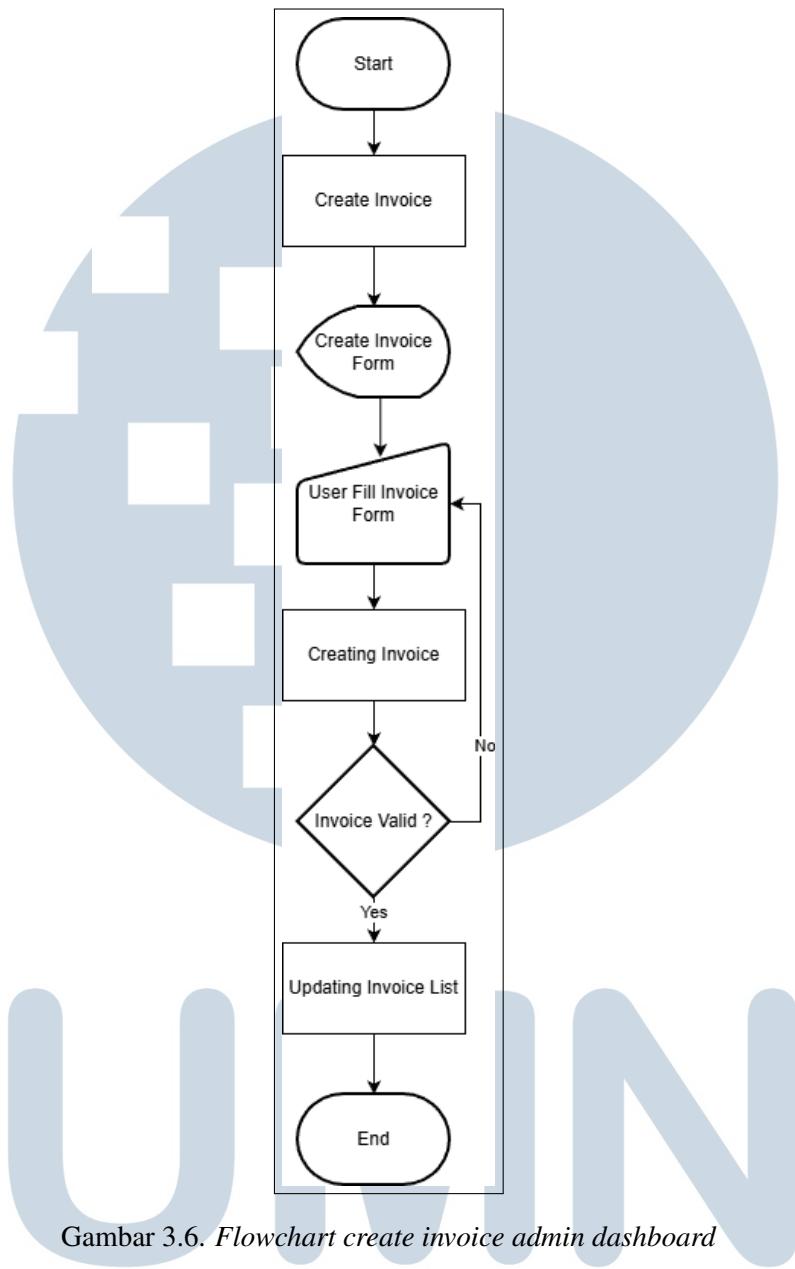
Gambar 3.5 merupakan *flowchart* yang menggambarkan alur pada *page invoice* untuk melihat *invoice detail* pada *Admin Dashboard*. Proses dimulai dari *Start*, kemudian sistem menampilkan halaman *Invoice* yang berisi *Invoice List* sebagai daftar *invoice* yang tersedia. Selanjutnya, pengguna memilih salah satu *invoice* dari daftar tersebut. Sistem kemudian mengecek keputusan *View Invoice Detail*. Jika pengguna memilih untuk melihat detail, maka sistem menampilkan

halaman *Invoice Detail* yang berisi informasi lengkap dari *invoice* yang terpilih, dan proses berakhir pada *End*. Apabila pengguna tidak memilih untuk melihat detail, alur akan kembali ke *Invoice List* sehingga pengguna tetap berada pada halaman daftar *invoice*.



Gambar 3.5. Flowchart invoice detail admin dashboard

Gambar 3.6 merupakan *flowchart* yang menjelaskan alur untuk *create invoice*, ketika *user* memilih *create invoice* maka *user* akan diarahkan ke *form invoice*, setelah *user* mengisi *form* tersebut akan dilakukan pengecekan apakah *invoice* yang baru dibuat adalah valid, Jika *invoice* tidak valid maka *user* akan diarahkan kembali untuk mengisi *form* ulang atau *user* bisa membatalkan pengisian *form invoice* dan kembali ke *page invoice*. Jika *invoice* valid maka *list invoice* akan di *update* dan memunculkan *invoice* yang baru di buat oleh *user*



Gambar 3.6. Flowchart create invoice admin dashboard

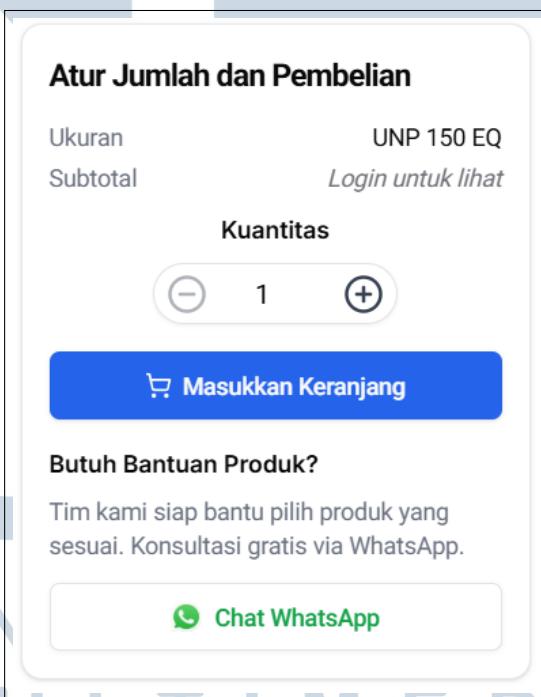
### 3.3.2 Implementasi

Website *e-commerce* PT Material Inovasi Industri telah diimplementasikan sesuai dengan perancangan sistem yang telah dibuat. Implementasi ini mencakup pengembangan antarmuka pengguna menggunakan framework Next.js dengan pendekatan *Server-Side Rendering* (SSR), sehingga halaman dapat dimuat secara optimal dan kontennya dapat diindeks dengan lebih baik oleh mesin pencari. Berikut ditampilkan beberapa halaman utama dari website *e-commerce* yang telah dikembangkan.

## A Halaman

berikut adalah beberapa halaman beserta potongan *code*-nya yang telah berhasil di implementasikan pada *website e-commerce* material inovasi industri.

Gambar 3.7 adalah Fitur *card* yang berfungsi sebagai komponen utama dalam proses pengaturan jumlah dan pembelian produk, yang menampilkan informasi ukuran produk serta subtotal harga yang hanya dapat diakses setelah pengguna melakukan *login*. Pengguna dapat menyesuaikan kuantitas produk melalui kontrol tombol *increment* dan *decrement* yang dirancang intuitif untuk menjaga validasi jumlah pembelian. Tombol aksi utama “*Masukkan Keranjang*” digunakan untuk menambahkan produk ke dalam keranjang belanja, sementara bagian bantuan di bawahnya menyediakan akses cepat ke layanan konsultasi melalui *WhatsApp*. Dengan desain yang ringkas dan informatif, *card* ini meningkatkan pengalaman pengguna dalam melakukan pembelian secara efisien dan terarah.



Gambar 3.7. Masukan keranjang

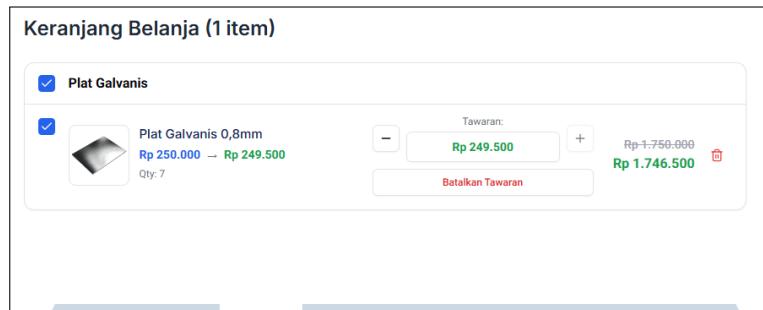
Kode 3.1 berfungsi sebagai *interface* untuk pengaturan jumlah produk dan proses penambahan ke keranjang belanja. Pengguna dapat menentukan jumlah pembelian melalui *Input* numerik yang dilengkapi dengan validasi batas minimum pembelian dan batas maksimum sesuai stok produk yang tersedia. Nilai kuantitas

yang dimasukkan akan digunakan sebagai dasar dalam proses penambahan produk ke keranjang. Tombol *Button* “*Masukkan Keranjang*” berfungsi untuk menjalankan aksi penambahan produk dan akan dinonaktifkan secara otomatis apabila data produk belum tersedia, stok produk habis, atau proses permintaan sedang berlangsung, sehingga dapat mencegah terjadinya kesalahan dalam proses pembelian.

```
1 <Input
2   type="number"
3   min="1"
4   max={selectedProductType?.data?.data?.stock || 999}
5   value={quantity}
6   onChange={(e) =>
7     handleQuantityChange(parseInt(e.target.value) || 1)
8   }
9 />
10
11 <Button
12   onClick={handleAddToCart}
13   disabled={
14     !selectedProductType?.data?.data ||
15     selectedProductType?.data?.data?.stock === 0 ||
16     addCartQuery.isPending
17   }
18 >
19   Masukkan Keranjang
20 </Button>
```

Kode 3.1: Implementasi fitur masukan keranjang

Gambar 3.8 menunjukkan halaman keranjang belanja yang berfungsi sebagai pusat pengelolaan transaksi sebelum proses pemesanan dilanjutkan ke tahap *checkout*. Halaman ini menampilkan daftar produk beserta informasi penting seperti nama produk, jumlah item, harga awal, dan harga hasil penawaran (*bid*). Pengguna dapat menyesuaikan jumlah produk, membatalkan penawaran, atau menghapus item dari keranjang, dengan total harga yang diperbarui secara dinamis. Fitur ini mendukung kejelasan transaksi, fleksibilitas pemesanan, dan peningkatan pengalaman pengguna (*User Experience*).



Gambar 3.8. Fitur *bidding*

Kode 3.2 merupakan potongan *code* implementasi fitur *bidding price* yang memungkinkan pengguna untuk mengubah dan membatalkan penawaran harga terhadap produk tertentu. Fitur ini menggunakan komponen *Dialog* sebagai media interaksi untuk menampilkan formulir pengeditan harga penawaran. Informasi yang ditampilkan meliputi harga asli produk dan harga penawaran saat ini, yang disajikan secara informatif untuk membantu pengguna dalam menentukan harga baru. Pengguna dapat memasukkan harga penawaran baru melalui komponen *Input*, yang kemudian diproses melalui fungsi penanganan perubahan dan pengiriman data. Selain itu, disediakan tombol *Button* untuk memperbarui penawaran maupun membatalkan proses, serta opsi *Batalkan Bid* yang memungkinkan pengguna menghapus penawaran harga secara langsung, sehingga mendukung fleksibilitas dalam proses negosiasi harga.

```

1 <Dialog open={ showBidDialog === item.id }>
2   <DialogTrigger asChild>
3     <Button size="sm">Edit Bid</Button>
4   </DialogTrigger>
5   <DialogContent>
6     <Input
7       placeholder="Harga penawaran baru"
8       value={bidPrices[item.id] || ''}
9       onChange={(e) =>
10         handleBidPriceChange(item.id, e.target.value)
11       }
12     />
13     <Button onClick={() => handleSubmitBid(item)}>
14       Update Penawaran
15     </Button>
16   </DialogContent>
17 </Dialog>
18

```

```

19 <Button onClick={() => handleRemoveBid(item.id)}>
20   Batalkan Bid
21 </Button>

```

Kode 3.2: Implementasi fitur *bidding price*

Gambar 3.9 merupakan bagian dari panel admin yang digunakan untuk mengelola seluruh data *invoice* yang terdapat pada aplikasi. Pada halaman ini, admin dapat melihat daftar *invoice* secara terstruktur dalam bentuk tabel yang menampilkan informasi penting seperti *Invoice ID*, *Status Pembayaran*, *Termin Pembayaran*, *Quotation ID*, *Customer*, *Total Order*, *Status Order*, *Tanggal Dibuat*, dan *Aktivitas Terakhir*. Selain itu, halaman ini dilengkapi dengan fitur pencarian, filter, dan pengaturan tampilan untuk memudahkan admin dalam menemukan data *invoice* tertentu. Admin juga dapat menambahkan *invoice* baru melalui tombol *create invoice*, serta mengelola data *invoice* yang sudah ada melalui menu aksi pada setiap baris, sehingga proses pemantauan dan pengelolaan transaksi dapat dilakukan secara lebih efektif dan efisien.

#	Invoice ID	Status	Termin Pembayaran	Quotation ID	Customer	Total Order	Status Order	Tgl. Dibuat	Aktivitas Terakhir	Action
1	INV-4207DRZQB	Unpaid	Lunas	ORD-I84UISR4W	Andreo	Rp 167.594	On-Process	11/11/2025, 00:17 AM	Nov 11, 2025	...
2	INV-5C1AHPPF3	Unpaid	Termin 30 Hari	ORD-Y5FN5KEX	AERO JACQUES TIMORENCE MANUKOA	Rp 994.448	On-Process	10/11/2025, 23:51 PM	Nov 10, 2025	...
3	INV-BIEL432KX	Paid	Lunas	ORD-E3QQ8XQFP	example	Rp 202.375	On-Process	10/11/2025, 23:42 PM	Nov 10, 2025	...
4	INV-UBUZJHSQ	Unpaid	Lunas	ORD-NF75HTDH	test	Rp 6.691.480	On-Process	10/11/2025, 23:41 PM	Nov 10, 2025	...
5	PKP Hidden	Paid	Lunas	ORD-I3NH5SV97	test	Rp 880.889	On-Process	08/11/2025, 13:56 PM	Nov 8, 2025	...
6	PKP Hidden	Unpaid	Lunas	ORD-NF75HTDH	-	Rp 0	-	07/11/2025, 13:30 PM	Nov 7, 2025	...
7	INV-ODDD3MT8L	Unpaid	Lunas	ORD-NF75HTDH	-	Rp 0	-	28/10/2025, 01:32 AM	Oct 28, 2025	...
8	PKP Hidden	Unpaid	Lunas	ORD-NF75HTDH	-	Rp 0	-	28/10/2025, 01:29 AM	Oct 28, 2025	...
9	INV-CM8XOD3QF	Unpaid	Lunas	ORD-NF75HTDH	-	Rp 0	-	28/10/2025, 01:29 AM	Oct 28, 2025	...
10	PKP Hidden	Unpaid	Lunas	ORD-R371HOL13	Eric	Rp 12.694.771	Pending	27/09/2025, 12:57 PM	Sep 27, 2025	...

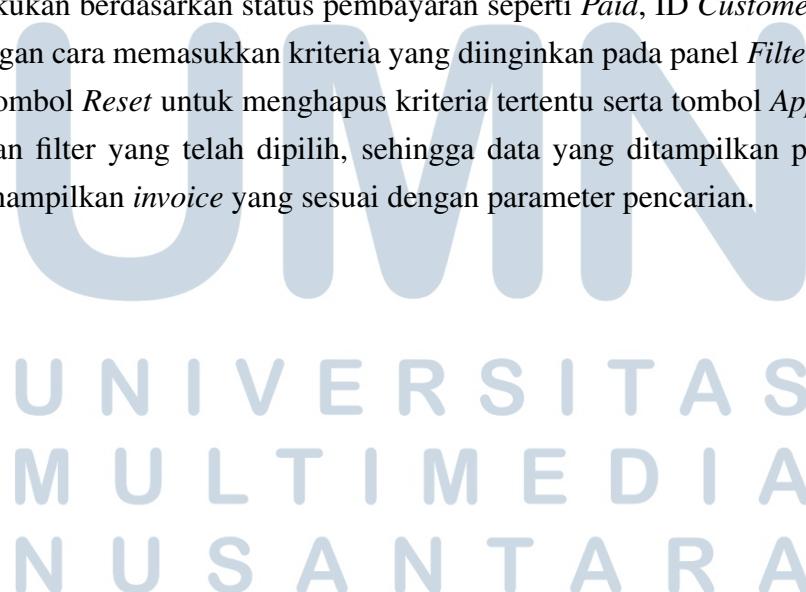
Gambar 3.9. Admin dashboard (*invoice list*)

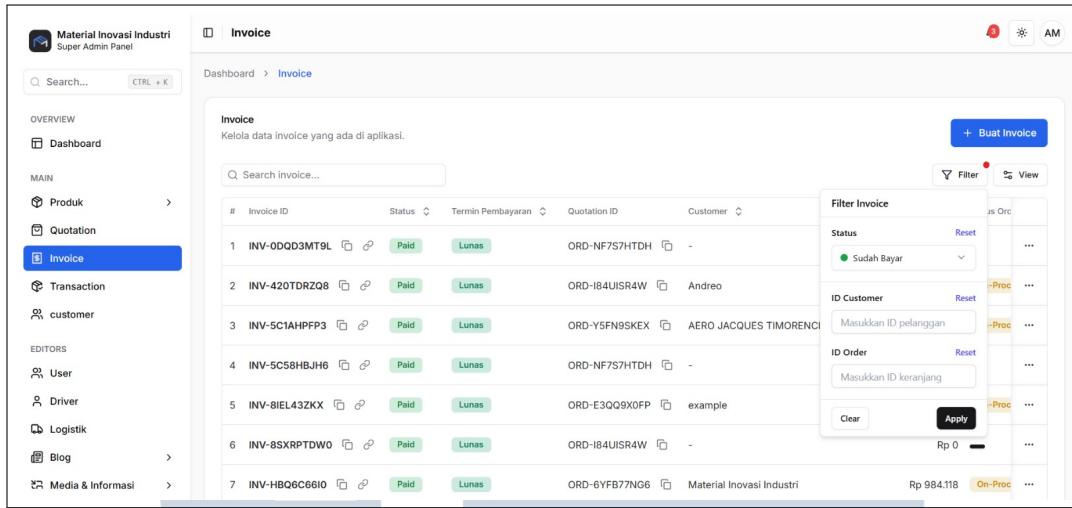
Kode 3.3 merupakan potongan *code* implementasi halaman *Admin Dashboard*. Halaman ini berfungsi untuk menampilkan daftar *invoice* secara terstruktur dalam bentuk tabel yang memuat informasi penting seperti *Invoice ID*, status pembayaran, dan data pelanggan. Selain itu, *admin dashboard* ini dilengkapi dengan fitur pencarian, pengaturan tampilan, serta penandaan status *invoice* menggunakan indikator visual untuk memudahkan admin dalam memantau dan mengelola data transaksi secara efektif dan efisien.

```
1 <table>
2   <tbody>
3     {invoices.map((invoice) => (
4       <tr key={invoice.id}>
5         <td>{ invoice.id }</td>
6         <td>{ invoice.invoiceId }</td>
7         <td>
8           <Badge className={getStatusColor(invoice.status)}>
9             { invoice.status }
10            </Badge>
11          </td>
12          <td>{ invoice.customer }</td>
13        </tr>
14      )) }
15    </tbody>
16  </table>
```

Kode 3.3: Potongan *code invoice list*

Gambar 3.10 merupakan fitur yang berfungsi untuk membantu admin menyaring data *invoice* agar lebih mudah ditemukan sesuai kebutuhan. Penyaringan dapat dilakukan berdasarkan status pembayaran seperti *Paid*, ID *Customer*, dan ID *Order* dengan cara memasukkan kriteria yang diinginkan pada panel *Filter Invoice*. Tersedia tombol *Reset* untuk menghapus kriteria tertentu serta tombol *Apply* untuk menerapkan filter yang telah dipilih, sehingga data yang ditampilkan pada tabel hanya menampilkan *invoice* yang sesuai dengan parameter pencarian.





Gambar 3.10. Admin dashboard (invoice filter)

Kode 3.4 ini merupakan potongan *code* komponen *client-side* pada *Next.js* yang berfungsi untuk menyaring data *invoice* berdasarkan beberapa kriteria, yaitu status, ID *Quotation*, dan ID *Customer*. Komponen ini memanfaatkan *state management* melalui *custom store* untuk menyimpan filter global serta *local state* (*tempFilters*) sebagai penampung sementara sebelum filter diterapkan. Antarmuka filter ditampilkan menggunakan komponen *Popover* dan *Select*, dilengkapi indikator visual berupa *badge* ketika filter aktif. Tombol *Apply* digunakan untuk menerapkan filter ke data utama, sedangkan tombol *Clear* dan *Reset* berfungsi untuk menghapus kriteria filter secara sebagian maupun keseluruhan.

```

1 const { filters, setFilters } = useInvoiceFilterStore();
2 const [tempFilters, setTempFilters] = useState(filters);
3
4 <Popover>
5   <PopoverTrigger asChild>
6     <Button size="sm" variant="outline">
7       <FilterIcon /> Filter
8     </Button>
9   </PopoverTrigger>
10
11 <PopoverContent>
12   <Select
13     value={tempFilters.status}
14     onValueChange={(value) =>
15       setTempFilters({ ...tempFilters, status: value })
16     }
17   >
```

```

18     <SelectItem value="Paid">Paid</SelectItem>
19     <SelectItem value="Unpaid">Unpaid</SelectItem>
20   </Select>
21
22   <Button onClick={() => setFilters(tempFilters)}>
23     Apply
24   </Button>
25 </PopoverContent>
26 </Popover>

```

Kode 3.4: Potongan *code* invoice filter

Gambar 3.11 merupakan halaman yang menampilkan informasi lengkap dari sebuah *invoice*, mulai dari identitas perusahaan penerbit, nomor *invoice*, status pembayaran, hingga tanggal penerbitan dan jatuh tempo. Pada bagian utama ditampilkan detail pengirim dan penerima, daftar item yang dipesan beserta kuantitas, harga, diskon, dan total per item, serta ringkasan perhitungan biaya yang mencakup *subtotal*, biaya pengiriman, biaya tambahan, dan total akhir pembayaran. Di sisi kanan halaman terdapat panel *Informasi Invoice* yang menyajikan data *order*, status proses, metode pengiriman, jumlah produk, serta *timeline* aktivitas untuk memudahkan pemantauan alur transaksi secara kronologis.

NUSANTARA  
Gambar 3.11. Admin dashboard (*invoice detail*)

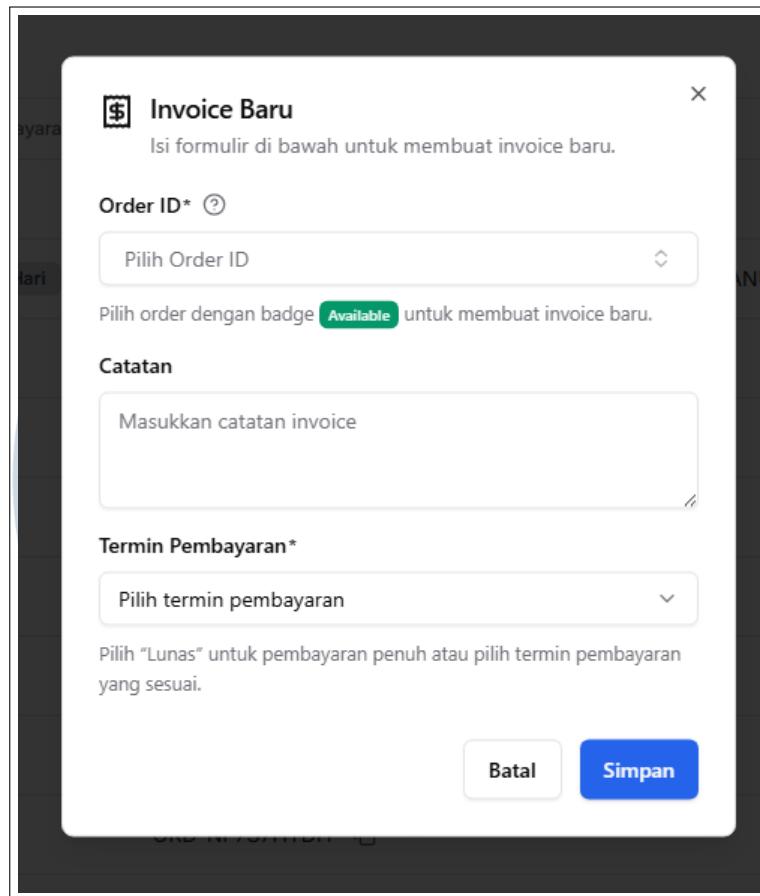
Kode 3.5 merupakan potongan *code* komponen *client-side* yang digunakan untuk menampilkan pratinjau detail *invoice* serta menyediakan fitur *print* dan ekspor ke *PDF*. Komponen ini mengambil data *invoice* dan profil perusahaan melalui

mekanisme *data fetching*, menampilkan daftar item dan total pembayaran dalam format A4, serta menyediakan panel aksi untuk mengelola *notes* dan status *invoice* seperti *Paid* dan *Cancel*.

```
1 const { data: invoice } = useQuery({
2   queryKey: ['invoice', invoiceId],
3   queryFn: () => invoiceServices.getInvoiceById({ id: invoiceId })
4   ,
5 }
6 <table>
7   <tbody>
8     {invoice?.data?.data?.order?.items?.map((item) => (
9       <tr key={item.id}>
10         <td>{item.product_type?.name}</td>
11         <td>{item.quantity}</td>
12         <td>{formatPriceIdr(item.price)}</td>
13         <td>{formatPriceIdr(item.price * item.quantity)}</td>
14       </tr>
15     )) }
16   </tbody>
17 </table>
18
19 <p>
20   Total: {formatPriceIdr(invoice?.data?.data?.order?.grand_total)}
21 </p>
```

Kode 3.5: Potongan *code* invoice detail

Gambar 3.12 merupakan fitur *create invoice* yang digunakan untuk membuat *invoice* baru dengan cara mengisi formulir yang tersedia, meliputi pemilihan *Order ID* yang berstatus *Available*, penambahan catatan *invoice*, serta penentuan termin pembayaran. Pengguna dapat memilih opsi *Lunas* untuk pembayaran penuh atau menentukan termin pembayaran sesuai kesepakatan, kemudian menyimpan data dengan menekan tombol *Simpan* atau membatalkan proses melalui tombol *Batal*, sehingga pembuatan *invoice* dapat dilakukan secara terstruktur dan terkontrol.



Gambar 3.12. Admin dashboard (create invoice)

Kode 3.6 merupakan potongan *code* komponen *client-side* yang menampilkan *form* untuk membuat *invoice* baru. Pengguna diwajibkan memilih *Order ID* yang tersedia, dengan bantuan *tooltip* dan *badge Available* sebagai indikator kelayakan order, serta dapat menambahkan catatan dan menentukan termin pembayaran. Formulir ini terintegrasi dengan *form handling* dan *validation*, dilengkapi tombol *Simpan* untuk memproses pembuatan *invoice* dan tombol *Batal* untuk membatalkan aksi, sehingga proses input data menjadi lebih aman, informatif, dan terkendali.

```
1 <Dialog open={isDialogOpen}>
2   <DialogTrigger asChild>
3     <Button>Buat Invoice</Button>
4   </DialogTrigger>
5
6   <DialogContent>
7     <form onSubmit={useFormInvoice.handleSubmit(onCreateInvoice)}>
8       <Select
```

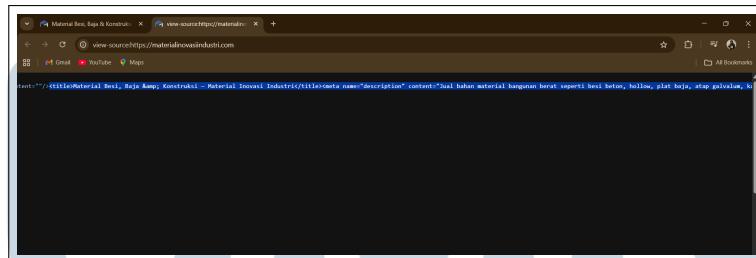
```

9     value={orderId}
10    onValueChange={(value) => setOrderId(value)}
11  />
12  <Textarea placeholder="Catatan invoice" />
13  <Select onValueChange={(value) => setPaymentTerm(value)}>
14    <SelectItem value="lunas">Lunas</SelectItem>
15    <SelectItem value="termin_30">Termin 30 Hari</SelectItem>
16  </Select>
17  <Button type="submit">Simpan</Button>
18  </form>
19  </DialogContent>
20</Dialog>

```

Kode 3.6: Implementasi *create invoice*

Gambar 3.13 menunjukkan implementasi optimasi mesin pencari (*SEO*) pada website *e-commerce* yang dikembangkan menggunakan *framework Next.js* dengan pendekatan *Server-Side Rendering (SSR)*. Melalui pendekatan ini, struktur HTML halaman beserta metadata seperti judul halaman (*title*) dan deskripsi (*meta description*) dihasilkan di sisi server sebelum halaman ditampilkan kepada pengguna, sehingga memungkinkan mesin pencari mengindeks konten halaman secara lebih efektif dibandingkan dengan pendekatan *Client-Side Rendering (CSR)*.



Gambar 3.13. Implementasi *SEO* website *e-commerce* material inovasi industri

### 3.4 Kendala dan Solusi yang Ditemukan

Selama magang sebagai *frontend developer* di PT Digital Go pada proyek website *e-commerce*, kendala yang dihadapi adalah data yang tidak konsisten dari *backend* yang memengaruhi proses pengembangan serta tantangan dalam memastikan tampilan tetap *responsive* di berbagai ukuran perangkat.

Untuk mengatasi kendala tersebut, digunakan *dummy data* selama tahap pengembangan agar komponen dapat dibangun tanpa bergantung pada kondisi *backend*. Komunikasi dengan tim *backend* dilakukan secara rutin untuk

menyesuaikan data dari *backend*. Pada sisi tampilan, pemanfaatan *Tailwind CSS* membantu mempercepat pembuatan tampilan *page* yang konsisten dan *responsive*.

