

Nama: Gigih Hermawan

NPM: 22312138

```
1 import { cookies } from "next/headers";
2
3 import { NextResponse } from "next/server";
4
5 export async function middleware(request) {
6   const cookieStore = await cookies();
7   const email = cookieStore.get("email")?.value;
8
9   if (
10     request.nextUrl.pathname.startsWith("/api/admin-dashboard/laptop") ||
11     request.nextUrl.pathname.startsWith("/admin-dashboard/laptop")
12   ) {
13     if (email !== "admin@gmail.com") {
14       return NextResponse.redirect(new URL("/", request.url));
15     }
16   }
17
18   if (request.nextUrl.pathname === "/admin-dashboard") {
19     if (email !== "admin@gmail.com") {
20       return NextResponse.redirect(new URL("/", request.url));
21     } else {
22       return NextResponse.redirect(
23         new URL("/admin-dashboard/laptop", request.url)
24       );
25     }
26   }
27
28   if (request.nextUrl.pathname === "/sign-in") {
29     if (email === "admin@gmail.com") {
30       return NextResponse.redirect(new URL("/", request.url));
31     }
32   }
33 }
34
```

Fitur:

Menambahkan middleware untuk melakukan pemrosesan di antara permintaan request dan response sebelum mencapai rute atau handler.

```

1 import { MongoClient } from "mongodb";
2
3 const uri = process.env.MONGODB_ATLAS_URI;
4 const client = new MongoClient(uri);
5
6 export async function GET(req) {
7   try {
8     await client.connect();
9     const database = client.db("e_katalog_penjualan_laptop");
10    const collection = database.collection("laptop");
11
12    const category = req?.nextUrl?.searchParams.get("category");
13    const search = req?.nextUrl?.searchParams.get("search");
14
15    let result;
16    if (category === "merk") {
17      result = await collection
18        .find({ merk: new RegExp(search, "i") })
19        .sort({ createdAt: -1 })
20        .toArray();
21    }
22    if (category === "nama") {
23      result = await collection
24        .find({ nama: new RegExp(search, "i") })
25        .sort({ createdAt: -1 })
26        .toArray();
27    }
28    if (category === "kapasitas_penyimpanan") {
29      result = await collection
30        .find({ kapasitas_penyimpanan: new RegExp(search, "i") })
31        .sort({ createdAt: -1 })
32        .toArray();
33    }
34    if (category === "ram") {
35      result = await collection
36        .find({ ram: new RegExp(search, "i") })
37        .sort({ createdAt: -1 })
38        .toArray();
39    }
40    if (category === "processor") {
41      result = await collection
42        .find({ processor: new RegExp(search, "i") })
43        .sort({ createdAt: -1 })
44        .toArray();
45    }
46    if (category === "vga") {
47      result = await collection
48        .find({ vga: new RegExp(search, "i") })
49        .sort({ createdAt: -1 })
50        .toArray();
51    }
52    if (category === "harga") {
53      result = await collection
54        .find({ harga: new RegExp(search, "i") })
55        .sort({ createdAt: -1 })
56        .toArray();
57    }
58
59    return new Response(
60      JSON.stringify({
61        messageResponse: "Data berhasil diambil dari Database",
62        result,
63      }),
64      {
65        status: 200,
66        statusText: "OK",
67      }
68    );
69  } catch (error) {
70    return new Response(error, {
71      status: 500,
72      statusText: "Internal Server Error",
73    });
74  } finally {
75    await client.close();
76  }
77 }
78

```

Fitur:

Menambahkan api endpoint untuk method GET pada halaman home.

```

1  import { cookies } from "next/headers";
2
3  export async function POST(req) {
4    try {
5      const cookieStore = await cookies();
6
7      const formData = await req.formData();
8      const email = formData.get("email");
9      const password = formData.get("password");
10
11     if (email !== "admin@gmail.com" || password !== "laptop1945") {
12       return new Response(
13         JSON.stringify({ messageResponse: "Invalid email or password." }),
14         {
15           status: 200,
16           statusText: "OK",
17         }
18       );
19     }
20
21     cookieStore.set("email", email);
22
23     return new Response(
24       JSON.stringify({ messageResponse: "Signed in successful." }),
25       {
26         status: 200,
27         statusText: "OK",
28       }
29     );
30   } catch (error) {
31     return new Response(
32       JSON.stringify({
33         messageResponse: "An error occurred, please try again later.",
34       }),
35       {
36         status: 500,
37         statusText: "Internal Server Error",
38       }
39     );
40   }
41 }
42

```

Fitur:

Menambahkan api endpoint untuk method POST pada halaman sign-in.



```

1  import { cookies } from "next/headers";
2
3  export async function POST() {
4    try {
5      const cookieStore = await cookies();
6      const email = cookieStore.get("email")?.value;
7
8      if (email) {
9        (await cookies()).delete("email");
10
11       return new Response(
12         JSON.stringify({ messageResponse: "Successfully signed out" }),
13         {
14           status: 200,
15           statusText: "OK",
16         }
17       );
18     } else {
19       return new Response(
20         JSON.stringify({ messageResponse: "No user is signed in" }),
21         {
22           status: 200,
23           statusText: "OK",
24         }
25       );
26     }
27   } catch (error) {
28     return new Response(
29       JSON.stringify({ messageResponse: "Error occurred during sign-out" }),
30       {
31         status: 500,
32         statusText: "Internal Server Error",
33       }
34     );
35   }
36 }
37

```

Fitur:

Menambahkan api endpoint untuk method POST pada komponen sign-out.

```
1 import { MongoClient, ObjectId } from "mongodb";
2
3 import { del, put } from "@vercel/blob";
4
5 const uri = process.env.MONGODB_ATLAS_URI;
6 const client = new MongoClient(uri);
7
8 export async function GET() {
9   try {
10     await client.connect();
11     const database = client.db("e_katalog_penyediaan_laptop");
12     const collection = database.collection("laptop");
13
14     const result = await collection.find({}).sort({ createdAt: -1 }).toArray();
15
16     return new Response(
17       JSON.stringify({
18         messageResponse: "Data berhasil diambil dari Database",
19         result,
20       }),
21       {
22         status: 200,
23         statusText: "OK",
24       }
25     );
26   } catch (error) {
27     return new Response(error, {
28       status: 500,
29       statusText: "Internal Server Error",
30     });
31   } finally {
32     await client.close();
33   }
34 }
35
```

Fitur:

Menambahkan api endpoint untuk method GET pada halaman admin-dashboard.

```

1 import { MongoClient, ObjectId } from "mongodb";
2
3 import { del, put } from "@vercel/blob";
4
5 const uri = process.env.MONGODB_ATLAS_URI;
6 const client = new MongoClient(uri);
7
8 export async function POST(req) {
9   try {
10     await client.connect();
11     const database = client.db("e_katalog_penjualan_laptop");
12     const collection = database.collection("laptop");
13
14     const formData = await req.formData();
15     const merk = formData.get("merk");
16     const nama = formData.get("nama");
17     const img_file = formData.get("img_file");
18
19     const img_file_buffer = new Uint8Array(await img_file.arrayBuffer());
20
21     const blob = await put(
22       `e-katalog-penjualan-laptop/${img_file.name}`,
23       img_file_buffer,
24       {
25         access: "public",
26       }
27     );
28
29     const kapasitas_penjualan = formData.get("kapasitas_penjualan");
30     const ram = formData.get("ram");
31     const processor = formData.get("processor");
32     const vga = formData.get("vga");
33     const harga = formData.get("harga");
34
35     const date = new Intl.DateTimeFormat("id-ID", {
36       dateStyle: "short",
37       timeStyle: "long",
38       timeZone: "Asia/Jakarta",
39     }).format(new Date());
40
41     await collection.insertOne({
42       merk,
43       nama,
44       img_file: {
45         pathname: blob.pathname,
46         url: blob.url,
47       },
48       kapasitas_penjualan,
49       ram,
50       processor,
51       vga,
52       harga,
53       createdAt: date,
54       updatedAt: date,
55     });
56
57     return new Response(
58       JSON.stringify({
59         messageResponse: "Data berhasil disimpan di Database",
60       }),
61       {
62         status: 200,
63         statusText: "OK",
64       }
65     );
66   } catch (error) {
67     return new Response(error, {
68       status: 500,
69       statusText: "Internal Server Error",
70     });
71   } finally {
72     await client.close();
73   }
74 }
75

```

Fitur:

Menambahkan api endpoint untuk method POST pada halaman admin-dashboard.

```

1  import { MongoClient, ObjectId } from "mongodb";
2
3  import { del, put } from "@vercel/blob";
4
5  const uri = process.env.MONGODB_ATLAS_URI;
6  const client = new MongoClient(uri);
7
8  export async function PUT(req) {
9    try {
10     await client.connect();
11     const database = client.db("e_katalog_penjualan_laptop");
12     const collection = database.collection("laptop");
13
14     const formData = await req.formData();
15     const _id = formData.get("_id");
16     const merk = formData.get("merk");
17     const nama = formData.get("nama");
18
19     const kapasitas_penjualan = formData.get("kapasitas_penjualan");
20     const ram = formData.get("ram");
21     const processor = formData.get("processor");
22     const vga = formData.get("vga");
23     const harga = formData.get("harga");
24
25     const date = new Intl.DateTimeFormat("id-ID", {
26       dateStyle: "short",
27       timeStyle: "long",
28       timeZone: "Asia/Jakarta",
29     }).format(new Date());
30
31     if (formData.has("img_file")) {
32       const img_file = formData.get("img_file");
33
34       const img_file_buffer = new Uint8Array(await img_file.arrayBuffer());
35
36       const blob = await put(
37         `e-katalog-penjualan-laptop/${img_file.name}`,
38         img_file_buffer,
39         {
40           access: "public",
41         }
42       );
43
44       await collection.updateOne(
45         { _id: new ObjectId(`${_id}`) },
46         {
47           $set: {
48             merk,
49             nama,
50             img_file: {
51               pathname: blob.pathname,
52               url: blob.url,
53             },
54             kapasitas_penjualan,
55             ram,
56             processor,
57             vga,
58             harga,
59             updatedAt: date,
60           },
61         }
62       );
63     } else {
64       await collection.updateOne(
65         { _id: new ObjectId(`${_id}`) },
66         {
67           $set: {
68             merk,
69             nama,
70             kapasitas_penjualan,
71             ram,
72             processor,
73             vga,
74             harga,
75             updatedAt: date,
76           },
77         }
78       );
79     }
80
81     return new Response(
82       JSON.stringify({
83         messageResponse: "Data berhasil diubah di Database",
84       }),
85       {
86         status: 200,
87         statusText: "OK",
88       }
89     );
90   } catch (error) {
91     return new Response(error, {
92       status: 500,
93       statusText: "Internal Server Error",
94     });
95   } finally {
96     await client.close();
97   }
98 }
99

```

Fitur:

Menambahkan api endpoint untuk method PUT pada halaman admin-dashboard.

```
1 import { MongoClient, ObjectId } from "mongodb";
2
3 import { del, put } from "@vercel/blob";
4
5 const uri = process.env.MONGODB_ATLAS_URI;
6 const client = new MongoClient(uri);
7
8 export async function DELETE(req) {
9   try {
10     await client.connect();
11     const database = client.db("e_katalog_penjualan_laptop");
12     const collection = database.collection("laptop");
13
14     const formData = await req.formData();
15     const _id = formData.get("_id");
16
17     const findDataById = await collection
18       .find({ _id: new ObjectId(`${_id}`) })
19       .toArray();
20
21     await del(findDataById[0].img_file.url);
22
23     await collection.deleteOne({ _id: new ObjectId(`${_id}`) });
24
25     return new Response(
26       JSON.stringify({
27         messageResponse: "Data berhasil dihapus di Database",
28       }),
29       {
30         status: 200,
31         statusText: "OK",
32       }
33     );
34   } catch (error) {
35     return new Response(error, {
36       status: 500,
37       statusText: "Internal Server Error",
38     });
39   } finally {
40     await client.close();
41   }
42 }
43
```

Fitur:

Menambahkan api endpoint untuk method DELETE pada halaman admin-dashboard.


```

1  const SearchLaptop = ({ router, category, setCategory, search, setSearch }) => {
2    return (
3      <div className="m-16 mb-8">
4        <h1 className="text-3xl font-medium text-center mb-8">Laptop Xmute</h1>
5
6        <form
7          className="flex flex-wrap gap-9 place-content-center mx-20"
8          onSubmit={(e) => {
9            e.preventDefault();
10
11            router.push(`/?category=${category}&search=${search}`);
12          }}
13        >
14          <div className="flex flex-wrap gap-3">
15            <select
16              className="bg-gray-200 p-2 border rounded border-black"
17              value={category}
18              onChange={(e) => {
19                setCategory(e.target.value);
20              }}
21            >
22              <option value="merk">Merk</option>
23              <option value="nama">Nama</option>
24              <option value="kapasitas_penyimpanan">Kapasitas Penyimpanan</option>
25              <option value="ram">RAM</option>
26              <option value="processor">Processor</option>
27              <option value="vga">VGA</option>
28              <option value="harga">Harga</option>
29            </select>
30
31            <input
32              className="bg-gray-200 w-80 p-2 border rounded border-black"
33              type="text"
34              search={search}
35              onChange={(e) => setSearch(e.target.value)}
36            />
37          </div>
38
39          <button
40            className="bg-gray-700 py-1 px-3 rounded text-white"
41            type="submit"
42          >
43            Search
44          </button>
45        </form>
46      </div>
47    );
48  };
49
50  export default SearchLaptop;
51

```

Fitur:

Menambahkan komponen cari laptop berdasarkan Kategori Merk, Nama, Kapasitas Penyimpanan, RAM, Processor, VGA, Harga.

```

1 import { MongoClient } from "mongodb";
2
3 const uri = process.env.MONGODB_ATLAS_URI;
4 const client = new MongoClient(uri);
5
6 export async function GET(req) {
7   try {
8     await client.connect();
9     const database = client.db("e_katalog_penjualan_laptop");
10    const collection = database.collection("laptop");
11
12    const category = req?.nextUrl?.searchParams.get("category");
13    const search = req?.nextUrl?.searchParams.get("search");
14
15    let result;
16    if (category === "merk") {
17      result = await collection
18        .find({ merk: new RegExp(search, "i") })
19        .sort({ createdAt: -1 })
20        .toArray();
21    }
22    if (category === "nama") {
23      result = await collection
24        .find({ nama: new RegExp(search, "i") })
25        .sort({ createdAt: -1 })
26        .toArray();
27    }
28    if (category === "kapasitas_penyimpanan") {
29      result = await collection
30        .find({ kapasitas_penyimpanan: new RegExp(search, "i") })
31        .sort({ createdAt: -1 })
32        .toArray();
33    }
34    if (category === "ram") {
35      result = await collection
36        .find({ ram: new RegExp(search, "i") })
37        .sort({ createdAt: -1 })
38        .toArray();
39    }
40    if (category === "processor") {
41      result = await collection
42        .find({ processor: new RegExp(search, "i") })
43        .sort({ createdAt: -1 })
44        .toArray();
45    }
46    if (category === "vga") {
47      result = await collection
48        .find({ vga: new RegExp(search, "i") })
49        .sort({ createdAt: -1 })
50        .toArray();
51    }
52    if (category === "harga") {
53      result = await collection
54        .find({ harga: new RegExp(search, "i") })
55        .sort({ createdAt: -1 })
56        .toArray();
57    }
58
59    return new Response(
60      JSON.stringify({
61        messageResponse: "Data berhasil diambil dari Database",
62        result,
63      }),
64      {
65        status: 200,
66        statusText: "OK",
67      }
68    );
69  } catch (error) {
70    return new Response(error, {
71      status: 500,
72      statusText: "Internal Server Error",
73    });
74  } finally {
75    await client.close();
76  }
77 }
78

```

Fitur:

Menambahkan api endpoint untuk method GET pada halaman home untuk pencarian laptop berdasarkan Kategori tertentu.