

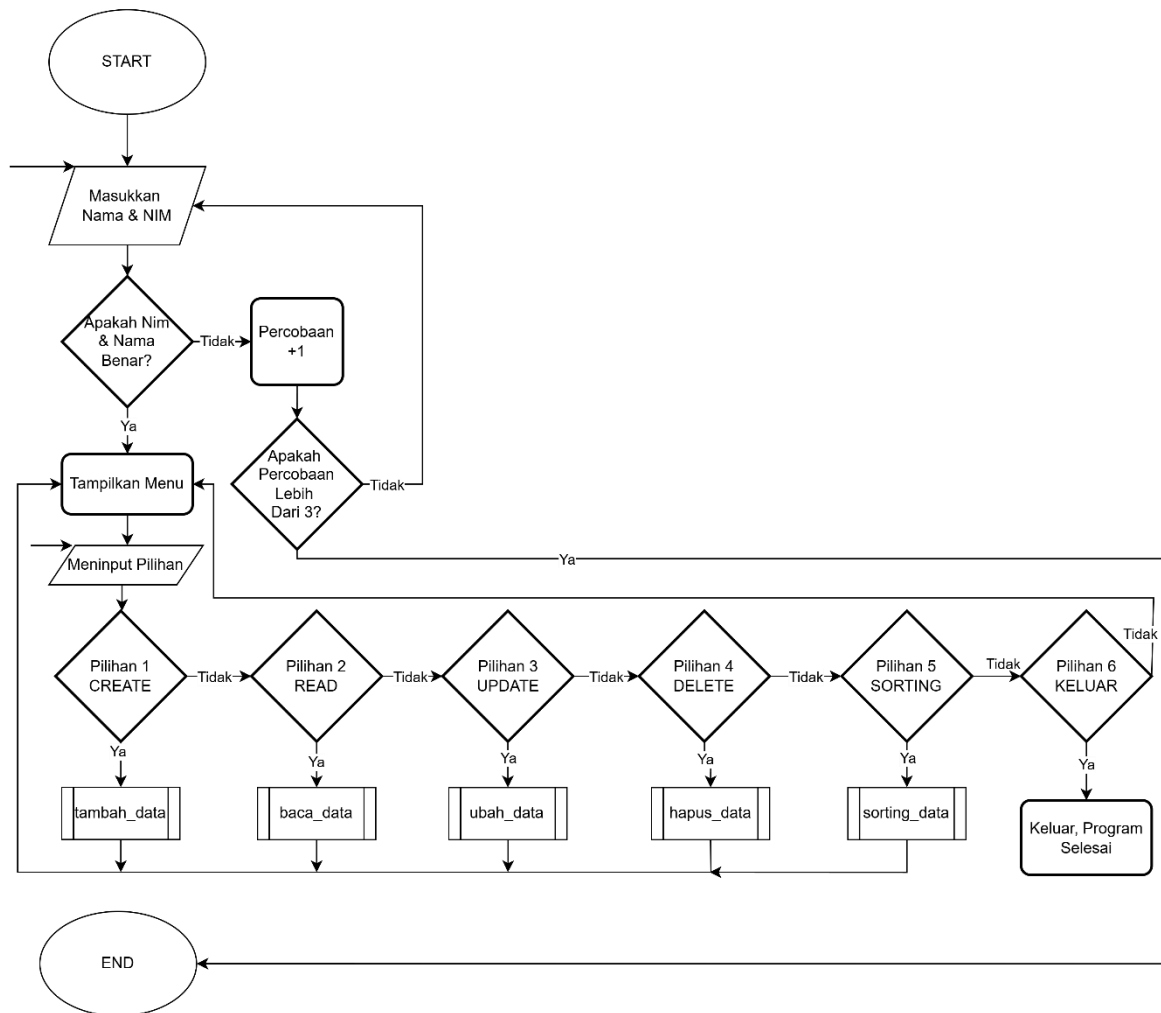
LAPORAN PRAKTIKUM
POSTTEST 6
ALGORITMA PEMROGRAMAN LANJUT



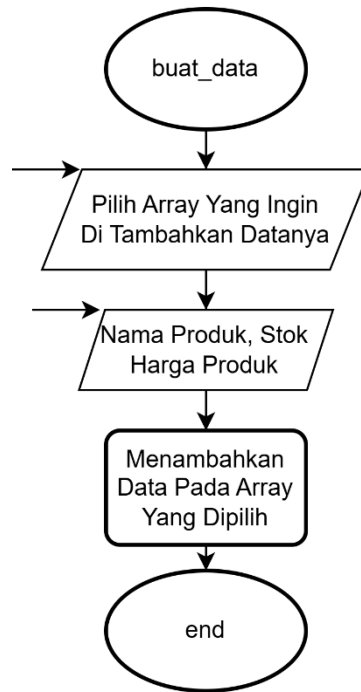
Disusun oleh:
Rangga Aditya Rahman (2409106044)
Kelas (A2 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

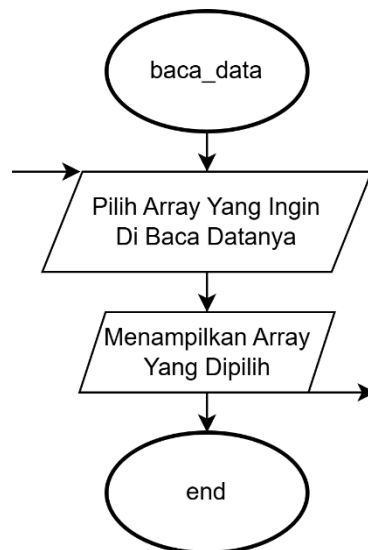
1. Flowchart



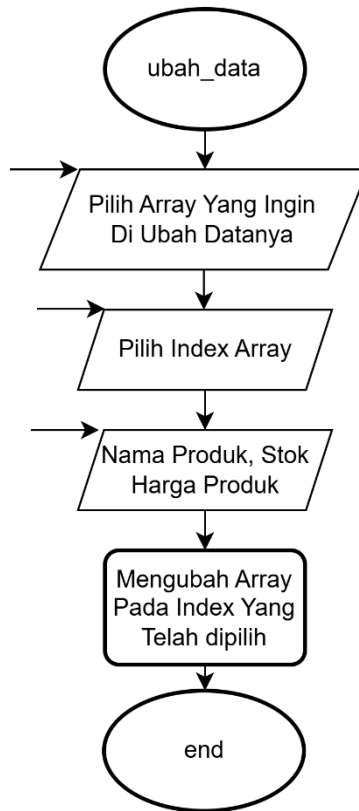
Gambar 1.1 Flowchart Program



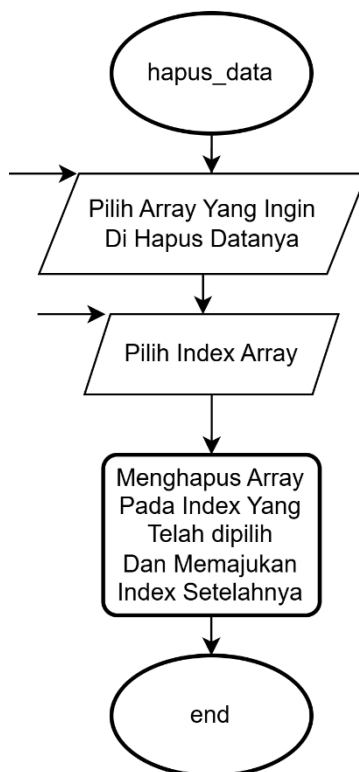
Gambar 1.2 Flowchart Fungsi Buat Data



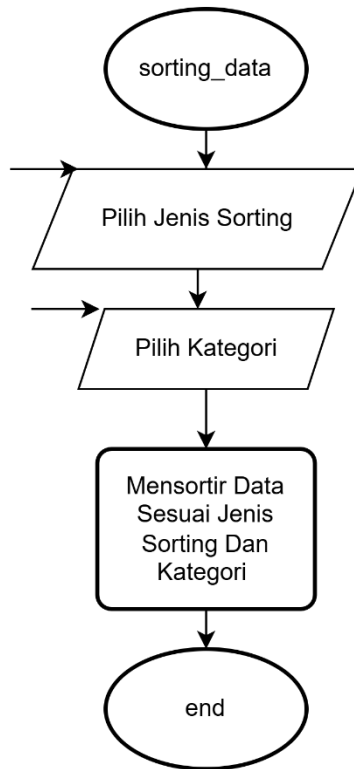
Gambar 1.3 Flowchart Fungsi Baca Data



Gambar 1.4 Flowchart Ubah Data



Gambar 1.5 Flowchart Hapus Data



Gambar 1.6 Flowchart Sorting Data

2. Analisis Program

2.1 Deskripsi Singkat Program

Program ini bertujuan untuk mengelola data barang dari tiga kategori utama, yaitu PlayStation, Xbox, dan Nintendo. Program ini memiliki fitur login dengan Batasan tiga kali percobaan serta fitur CRUD (Create, Read, Update, Delete) untuk mengelola data barang dalam setiap kategori. Di Posttest kali ini data barang menggunakan Pointer dan Sorting. Sorting yang dipake ada 3 yaitu 1. Insertion Sort (Desc), 2. Quick Sort (Asc) dan 3. Selection Sort (Asc)

3. Source Code

A. Login

Fitur ini digunakan pengguna untuk login menggunakan akun yang sudah ada, yaitu dengan username Rangga dan password NIM 044.

Source Code:

```
while (percobaan <= 3) {
    cout << "Percobaan ke: " << percobaan << endl;
    cout << "Masukkan Nama: ";
    cin >> namacek;
    cout << "Masukkan Password (NIM): ";
    cin >> nimcek;

    if (namacek == nama && nimcek == nim) {
        cout << "Login Berhasil!" << endl;
        break;
    } else {
        cout << "Percobaan Gagal!" << endl;
        percobaan++;
        cin.clear();
        cin.ignore(10000, '\n');
    }
}
if (percobaan > 3){
    cout << "\nLogin Gagal";
    return 0;
}
```

B. Menambah Data Array (CREATE)

Fitur Ini Menambahkan Data Ke Dalam Array Sesuai Kategori Array Yang Dipilih.

Source Code:

```
void tambah_data(int kategori)
{
    if (kategori == 1)
    {
        cout << "\n";
        cin.ignore();
        if (indexps < MAX_ARRAY)
        {
            cout << "Masukkan nama item : ";
        }
    }
}
```

```

        getline(cin, playstation[indexps].nama);
        cout << "Masukkan stock item : ";
        getline(cin, playstation[indexps].stock);
        cout << "Masukkan harga item : ";
        getline(cin, playstation[indexps].harga);
        indexps++;
    }
    else
    {
        cout << "Data Array Penuh!!";
    }
}

else if (kategori == 2)
{
    cout << "\n";
    cin.ignore();
    if (indexxbox < MAX_ARRAY)
    {
        cout << "Masukkan nama item : ";
        getline(cin, xbox[indexxbox].nama);
        cout << "Masukkan stock item : ";
        getline(cin, xbox[indexxbox].stock);
        cout << "Masukkan harga item : ";
        getline(cin, xbox[indexxbox].harga);
        indexxbox++;
    }
    else
    {
        cout << "Data Array Penuh!!";
    }
}

else if (kategori == 3)
{
    cout << "\n";
    cin.ignore();
    if (indexnintendo < MAX_ARRAY)
    {
        cout << "Masukkan nama item : ";
        getline(cin, nintendo[indexnintendo].nama);
        cout << "Masukkan stock item : ";
        getline(cin, nintendo[indexnintendo].stock);
        cout << "Masukkan harga item : ";
        getline(cin, nintendo[indexnintendo].harga);
        indexnintendo++;
    }
    else
    {
        cout << "Data Array Penuh!!";
    }
}

```

```

    }
    else
    {
        cout << "Pilihan tidak valid! Coba lagi." << endl;
    }
}

```

C. Membaca Data Array (READ)

Bagian ini adalah bagian membaca/menampilkan array ke terminal sesuai kategori array yang dipilih.

Source Code:

```

void baca_data(int kategori)
{
    if (kategori == 1)
    {
        cout << "\n=====\\n";
        cout << "| No | Nama Produk | Stok | Harga |\\n";
        cout << "=====\\n";
        for (int i = 0; i < indexps; i++)
        {
            cout << "| " << i + 1 << " | "
                << playstation[i].nama << " | "
                << playstation[i].stock << " | "
                << playstation[i].harga << " |\\n";
        }
        cout << "=====\\n";
    }
    else if (kategori == 2)
    {
        cout << "\n=====\\n";
        cout << "| No | Nama Produk | Stok | Harga |\\n";
        cout << "=====\\n";
        for (int i = 0; i < indexxbox; i++)
        {
            cout << "| " << i + 1 << " | "
                << xbox[i].nama << " | "
                << xbox[i].stock << " | "
                << xbox[i].harga << " |\\n";
        }
        cout << "=====\\n";
    }
    else if (kategori == 3)

```



```

{
    cout << "\n=====\\n";
    cout << "| No | Nama Produk | Stok | Harga |\\n";
    cout << "=====\\n";
    for (int i = 0; i < indexnintendo; i++)
    {
        cout << "| " << i + 1 << ". | "
              << nintendo[i].nama << " | "
              << nintendo[i].stock << " | "
              << nintendo[i].harga << " |\\n";
    }
    cout << "=====\\n";
}
else
{
    cout << "Pilihan tidak valid! Coba lagi.\\n";
}
}

```

D. Mengubah Data Array (UPDATE)

Bagian Ini Adalah Bagian Mengubah Data Array Sesuai Index Yang Dipilih Dan Juga Kategori Yang Di Pilih Dan Disini Saya Sedikit Memangkas Codingan Dengan Pointer

Source Code :

```

void ubah_data_pointer(konsolgame* item)
{
    cout << "Masukkan nama item : ";
    getline(cin, item->nama);
    cout << "Masukkan stock item : ";
    getline(cin, item->stock);
    cout << "Masukkan harga item : ";
    getline(cin, item->harga);
}

```

```

// Perubahan menggunakan pointer untuk menghemat codingan
=====

```

```

void ubah_data(int kategori)
{
    if (kategori == 1)
    { // Playstation Update
        cout << "\n";
        cin.ignore();
        cout << "Nomor | Nama Barang | Stock Barang | Harga Barang" << endl;

        if (indexps == 0)
        {
            cout << "Data Playstation kosong!\n";
        }

        for (int i = 0; i < indexps; i++)
        {
            cout << i + 1 << ". " << playstation[i].nama << " " <<
playstation[i].stock << " " << playstation[i].harga << endl;
        }

        cout << "Pilihlah Nomor Data Yang Ingin Diubah : ";
        cin >> pilihanupdate;
        cin.ignore();
        pilihanupdate--;
        if (pilihanupdate < 0 || pilihanupdate >= indexps)
        {
            cout << "Pilihan tidak valid! Silakan coba lagi.\n";
        }
        ubah_data_pointer(&playstation[pilihanupdate]);
    }

    else if (kategori == 2)
    { // Xbox Update
        cout << "\n";
        cin.ignore();
        cout << "Nomor | Nama Barang | Stock Barang | Harga Barang" << endl;

        if (indexxbox == 0)
        {
            cout << "Data Xbox kosong!\n";
        }

        for (int i = 0; i < indexxbox; i++)
        {
            cout << i + 1 << ". " << xbox[i].nama << " " << xbox[i].stock << " "
<< xbox[i].harga << endl;
        }

        cout << "Pilihlah Nomor Data Yang Ingin Diubah!! ";
        cin >> pilihanupdate;
        pilihanupdate--;
    }
}

```

```

        cin.ignore();

        if (pilihanupdate < 0 || pilihanupdate >= indexxbox)
        {
        }
        ubah_data_pointer(&xbox[pilihanupdate]);
    }

    else if (kategori == 3)
    { // Nintendo Update
        cout << "\n";
        cin.ignore();
        cout << "Nomor | Nama Barang | Stock Barang | Harga Barang" << endl;

        if (indexnintendo == 0)
        {
            cout << "Data Nintendo kosong!\n";
        }

        for (int i = 0; i < indexnintendo; i++)
        {
            cout << i + 1 << ". " << nintendo[i].nama << " " <<
nintendo[i].stock << " " << nintendo[i].harga << endl;
        }

        cout << "Pilihlah Data Yang Ingin Diubah!";
        cin >> pilihanupdate;
        pilihanupdate--;
        cin.ignore();

        if (pilihanupdate < 0 || pilihanupdate >= indexnintendo)
        {
            cout << "Pilihan tidak valid! Silakan coba lagi.\n";
        }

        ubah_data_pointer(&nintendo[pilihanupdate]);
    }

    else
    {
        cout << "Pilihan tidak valid! Coba lagi." << endl;
    }
}
//
=====
=====

```

E. Menghapus Data Array (DELETE)

Bagian Ini Adalah Bagian Menghapus Index Yang Dipilih Pada Array Dan Index Setelahnya Naik Mengisi Index Yang Terhapus.

Source Code :

```
void hapus_data(int kategori)
{
    if (kategori == 1)
    { // Hapus Playstation
        cout << "\n";
        cin.ignore();
        cout << "Nomor | Nama Barang | Stock Barang | Harga Barang" << endl;

        if (indexps == 0)
        {
            cout << "Data Playstation kosong!\n";
        }

        for (int i = 0; i < indexps; i++)
        {
            cout << i + 1 << ". " << playstation[i].nama << " " <<
            playstation[i].stock << " " << playstation[i].harga << endl;
        }

        cout << "Pilihlah Nomor Data Yang Ingin Dihapus : ";
        cin >> pilihanhapus;

        if (pilihanhapus < 1 || pilihanhapus > indexps)
        {
            cout << "Pilihan tidak valid!\n";
        }

        for (int i = pilihanhapus - 1; i < indexps - 1; i++)
        {
            playstation[i].nama = playstation[i + 1].nama;
            playstation[i].stock = playstation[i + 1].stock;
            playstation[i].harga = playstation[i + 1].harga;
        }
        indexps--;
    }
    else if (kategori == 2)
    { // Hapus Xbox
        cout << "\n";
        cin.ignore();
        cout << "Nomor | Nama Barang | Stock Barang | Harga Barang" << endl;

        if (indexxbox == 0)
```

```

    {
        cout << "Data Xbox kosong!\n";
    }

    for (int i = 0; i < indexxbox; i++)
    {
        cout << i + 1 << ". " << xbox[i].nama << " " << xbox[i].harga << " "
        << xbox[i].stock << endl;
    }

    cout << "Pilihlah Nomor Data Yang Ingin Dihapus : ";
    cin >> pilihanhapus;

    if (pilihanhapus < 1 || pilihanhapus > indexxbox)
    {
        cout << "Pilihan tidak valid!\n";
    }

    for (int i = pilihanhapus - 1; i < indexxbox - 1; i++)
    {
        xbox[i].nama = xbox[i + 1].nama;
        xbox[i].stock = xbox[i + 1].stock;
        xbox[i].harga = xbox[i + 1].harga;
    }
    indexxbox--;
}
else if (kategori == 3)
{ // Hapus Nintendo
    cout << "\n";
    cin.ignore();
    cout << "Nomor | Nama Barang | Stock Barang | Harga Barang" << endl;

    if (indexnintendo == 0)
    {
        cout << "Data Nintendo kosong!\n";
    }

    for (int i = 0; i < indexnintendo; i++)
    {
        cout << i + 1 << ". " << nintendo[i].nama << " " <<
nintendo[i].stock << " " << nintendo[i].harga << endl;
    }

    cout << "Pilihlah Nomor Data Yang Ingin Dihapus : ";
    cin >> pilihanhapus;

    if (pilihanhapus < 1 || pilihanhapus > indexnintendo)
    {
        cout << "Pilihan tidak valid!\n";
    }
}

```

```

        for (int i = pilihanhapus - 1; i < indexnintendo - 1; i++)
        {
            nintendo[i].nama = nintendo[i + 1].nama;
            nintendo[i].stock = nintendo[i + 1].stock;
            nintendo[i].harga = nintendo[i + 1].harga;
        }
        indexnintendo--;
    }
    else
    {
        cout << "Pilihan kategori tidak valid!\n";
    }
}

```

F. Deklarasi Variabel

Bagian Ini Adalah Codingan Berisikan Deklarasi Variabel Yang Digunakan Pada Program

Source Code:

```

string nama = "Rangga", namacek;
int nim = 44, percobaan = 0, nimcek, pilihan, index;
struct konsolgame
{
    string nama;
    string stock;
    string harga;
};

konsolgame playstation[MAX_ARRAY];
konsolgame xbox[MAX_ARRAY];
konsolgame nintendo[MAX_ARRAY];

int indexps = 0, indexxbox = 0, indexnintendo = 0;

```

G. Sorting

Bagian Ini Adalah Codingan Yang Berisikan Condingan Sorting Untuk Menjalankan Mensortir Data Secara Ascending Atau Descending.

Source Code:

```

void insertionSortDesc(konsolgame *data, const int &jumlah)
{

```

```

    for (int i = 1; i < jumlah; i++)
    {
        konsolgame key = data[i];
        int j = i - 1;
        while (j >= 0 && data[j].nama < key.nama)
        {
            data[j + 1] = data[j];
            j--;
        }
        data[j + 1] = key;
        cout << "Tahap ke-" << i << ": ";
        for (int k = 0; k < jumlah; k++)
        {
            cout << data[k].nama << " ";
        }
        cout << endl;
    }
}

void quickSortAsc(konsolgame *data, int low, int high)
{
    if (low >= high)
        return;
    konsolgame pivot = data[low + (high - low) / 2];
    int i = low, j = high;

    while (i <= j)
    {
        while (stoi(data[i].stock) < stoi(pivot.stock))
        {
            i++;
        }
        while (stoi(data[j].stock) > stoi(pivot.stock))
        {
            j--;
        }
        if (i <= j)
        {
            swap(data[i], data[j]);
            i++;
            j--;
        }
    }
    if (low < j)
    {
        quickSortAsc(data, low, j);
    }
    if (i < high)
    {
        quickSortAsc(data, i, high);
    }
}

```

```

}

void selectionSortAsc(konsolgame *data, int panjang)
{
    for (int i = 0; i < panjang - 1; i++)
    {
        int minIndex = i;
        for (int j = i + 1; j < panjang; j++)
        {
            if (stoi(data[j].harga) < stoi(data[minIndex].harga))
            {
                minIndex = j;
            }
        }
        if (minIndex != i)
        {
            swap(data[i], data[minIndex]);
        }
        cout << "Tahap ke-" << i + 1 << ": ";
        for (int k = 0; k < panjang; k++)
        {
            cout << data[k].harga << " ";
        }
        cout << endl;
    }
}

void sorting_data(int pilihsorting) {
    int pilihan;
    cout << "\nPilih Kategori:\n";
    cout << "1. PlayStation\n2. Xbox\n3. Nintendo\n";
    cout << "Masukkan pilihan: ";
    cin >> pilihan;

    if (pilihan == 1) {
        if (pilihsorting == 1) {
            insertionSortDesc(playstation, indexps);
        } else if (pilihsorting == 2) {
            quickSortAsc(playstation, 0, indexps - 1);
        } else if (pilihsorting == 3) {
            selectionSortAsc(playstation, indexps);
        }
    }

    } else if (pilihan == 2) {
        if (pilihsorting == 1) {
            insertionSortDesc(xbox, indexxbox);
        } else if (pilihsorting == 2) {
            quickSortAsc(xbox, 0, indexxbox - 1);
        } else if (pilihsorting == 3) {
            selectionSortAsc(xbox, indexxbox);
        }
    }
}

```



```

    } else if (pilihan == 3) {
        if (pilihsorting == 1) {
            insertionSortDesc(nintendo, indexnintendo);
        } else if (pilihsorting == 2) {
            quickSortAsc(nintendo, 0, indexnintendo - 1);
        } else if (pilihsorting == 3) {
            selectionSortAsc(nintendo, indexnintendo);
        }
    }
}
}

```

F. Main Program

Bagian Ini Adalah Codingan Yang Berisikan Condingan Inti Untuk Menjalankan Program Ini

Source Code:

```

int main()
{
    while (percobaan <= 3)
    {
        cout << "Percobaan ke: " << percobaan << endl;
        cout << "Masukkan Nama: ";
        cin >> namacek;
        cout << "Masukkan Password (NIM): ";
        cin >> nimcek;

        if (namacek == nama && nimcek == nim)
        {
            cout << "Login Berhasil!" << endl;
            break;
        }
        else
        {
            cout << "Percobaan Gagal!" << endl;
            percobaan++;
            cin.clear();
            cin.ignore(10000, '\n');
        }
    }
    if (percobaan > 3)
    {
        cout << "\nLogin Gagal";
        return 0;
    }
}

```

```

do
{
    cout << "\n===== " << endl;
    cout << "          MENU UTAMA          " << endl;
    cout << "===== " << endl;
    cout << "1. Menambahkan Data" << endl;
    cout << "2. Membaca Array" << endl;
    cout << "3. Mengubah Array" << endl;
    cout << "4. Menghapus Array" << endl;
    cout << "5. Sorting Data" << endl;
    cout << "6. Keluar" << endl;
    cout << "===== " << endl;
    cout << "Pilih menu: ";
    cin >> pilihan;

    int kategori;

    switch (pilihan)
    {
        case 1: // Membuat
            cout << "\nPilih Kategori:\n";
            cout << "1. PlayStation\n2. Xbox\n3. Nintendo\n";
            cout << "Masukkan pilihan: ";
            cin >> kategori;
            tambah_data(kategori);
            break;
        case 2: // Membaca
            cout << "\nPilih Kategori:\n";
            cout << "1. PlayStation\n2. Xbox\n3. Nintendo\n";
            cout << "Masukkan pilihan: ";
            cin >> kategori;
            baca_data(kategori);
            break;
        case 3: // Update
            cout << "\nPilih Kategori:\n";
            cout << "1. PlayStation\n2. Xbox\n3. Nintendo\n";
            cout << "Masukkan pilihan: ";
            cin >> kategori;
            ubah_data(kategori);
            break;
        case 4: // Menghapus
            int pilihanhapus;
            cout << "\nPilih Kategori:\n";
            cout << "1. PlayStation\n2. Xbox\n3. Nintendo\n";
            cout << "Masukkan pilihan: ";
            cin >> kategori;
            hapus_data(kategori);
            break;
    }
}

```

```

        case 5:
            cout << "\nPilih Jenis Sorting:\n";
            cout << "1.Insertion Sort (Nama Menu Secara Descending)\n 2.Quick
Sort (Stock Barang Secara Ascending)\n 3.Selection Sort (Harga Barang Secara
Ascending)\n";
            cout << "Masukkan Pilihan: ";
            cin >> pilihsorting;
            sorting_data(pilihsorting);
            break;

        case 6:
            cout << "Terima kasih telah menggunakan program ini!" << endl;
            break;

        default:
            cout << "Pilihan tidak valid! Coba lagi." << endl;
            cin.clear();
            cin.ignore(10000, '\n');
    }

    } while (pilihan != 6);
}

```

4. Uji Coba dan Hasil Output

4.1 Hasil Output

```

Percobaan ke: 1
Masukkan Nama: ter
Masukkan Password (NIM): gd
Percobaan Gagal!
Percobaan ke: 2
Masukkan Nama: t43
Masukkan Password (NIM): fgd
Percobaan Gagal!
Percobaan ke: 3
Masukkan Nama: 532
Masukkan Password (NIM): t
Percobaan Gagal!
Login Gagal, kesempatan habis!

```

Gambar 4.1 Gagal Login

```

Percobaan ke: 0
Masukkan Nama: Rangga
Masukkan Password (NIM): 044
Login Berhasil!

=====
MENU UTAMA
=====
1. Menambahkan Data
2. Membaca Array
3. Mengubah Array
4. Menghapus Array
5. Keluar
=====
Pilih menu: 1

Pilih Kategori:
1. PlayStation
2. Xbox
3. Nintendo
Masukkan pilihan: 1

Masukkan nama item : ps1
Masukkan stock item : 10
Masukkan harga item : 1000000

```

Gambar 4.2 Login Berhasil Dan Menambahkan Data Ke Array Playstation

```

=====
MENU UTAMA
=====
1. Menambahkan Data
2. Membaca Array
3. Mengubah Array
4. Menghapus Array
5. Keluar
=====
Pilih menu: 2

Pilih Kategori:
1. PlayStation
2. Xbox
3. Nintendo
Masukkan pilihan: 1

=====
| No | Nama Produk | Stok | Harga |
=====
| 1 | ps1 | 10 | 1000000 |
=====

```

Gambar 4.3 Menampilkan Data Untuk Mengecek Array Playstation

```

=====
MENU UTAMA
=====
1. Menambahkan Data
2. Membaca Array
3. Mengubah Array
4. Menghapus Array
5. Keluar
=====
Pilih menu: 3

Pilih Kategori:
1. PlayStation
2. Xbox
3. Nintendo
Masukkan pilihan: 1

Nomor | Nama Barang | Stock Barang | Harga Barang
1. ps1 10 1000000
Pilihlah Nomor Data Yang Ingin Diubah : 1
Masukkan nama item : ps2
Masukkan stock item : 5
Masukkan harga item : 200000000

```

Gambar 4.4 Mengubah Array Playstation Sesuai Index Yang Dipilih

```

=====
MENU UTAMA
=====
1. Menambahkan Data
2. Membaca Array
3. Mengubah Array
4. Menghapus Array
5. Keluar
=====
Pilih menu: 2

Pilih Kategori:
1. PlayStation
2. Xbox
3. Nintendo
Masukkan pilihan: 1

=====
| No | Nama Produk | Stok | Harga |
=====
| 1 | ps2 | 5 | 200000000 |
=====

```

Gambar 4.5 Menampilkan Array Yang Udh Diubah Pada Array Playstation

```

=====
                        MENU UTAMA
=====
1. Menambahkan Data
2. Membaca Array
3. Mengubah Array
4. Menghapus Array
5. Keluar
=====
Pilih menu: 4

Pilih Kategori:
1. PlayStation
2. Xbox
3. Nintendo
Masukkan pilihan: 1

Nomor | Nama Barang | Stock Barang | Harga Barang
1. ps2 5 200000000
Pilihlah Nomor Data Yang Ingin Dihapus : 1

```

Gambar 4.6 Menghapus Index Pada Array Playstation

```

=====
                        MENU UTAMA
=====
1. Menambahkan Data
2. Membaca Array
3. Mengubah Array
4. Menghapus Array
5. Keluar
=====
Pilih menu: 2

Pilih Kategori:
1. PlayStation
2. Xbox
3. Nintendo
Masukkan pilihan: 1

=====
| No | Nama Produk | Stok | Harga |
=====
=====

=====
                        MENU UTAMA
=====
1. Menambahkan Data
2. Membaca Array
3. Mengubah Array
4. Menghapus Array
5. Keluar
=====
Pilih menu: 5
Terima kasih telah menggunakan program ini!

```

Gambar 4.7 Mengecek Yang Sudah Terhapus

No	Nama Produk	Stok	Harga
1	ps2	5	200000000
2	ps3	20	500000000
3	ps1	200	5000000

Gambar 4.8 Data Sebelum Disortir

No	Nama Produk	Stok	Harga
1	ps3	20	500000000
2	ps2	5	200000000
3	ps1	200	5000000

Gambar 4.9 Setelah Disortir Menggunakan Insertion Sort (Desc)

No	Nama Produk	Stok	Harga
1	ps2	5	200000000
2	ps3	20	500000000
3	ps1	200	5000000

Gambar 4.10 Setelah Disortir Menggunakan Quick Sort (Asc)

No	Nama Produk	Stok	Harga
1	ps1	200	5000000
2	ps2	5	200000000
3	ps3	20	500000000

Gambar 4.11 Setelah Disortir Menggunakan Selection Sort (Asc)

5. Git

```
PS C:\Users\USER\Documents\Tugas Kuliah\codingan\C++\pratikum-apl> git add .
warning: in the working copy of 'post-test/post-test-apl-6/2409106044-RanggaAdityaRahman-PT-6.cpp', LF will be replaced by
CRLF the next time Git touches it
PS C:\Users\USER\Documents\Tugas Kuliah\codingan\C++\pratikum-apl> git commit -m "posttest-6"
[master c19c9d9] posttest-6
2 files changed, 574 insertions(+)
create mode 100644 post-test/post-test-apl-6/2409106044-RanggaAdityaRahman-PT-6.cpp
create mode 100644 post-test/post-test-apl-6/2409106044-RanggaAdityaRahman-PT-6.exe
PS C:\Users\USER\Documents\Tugas Kuliah\codingan\C++\pratikum-apl> git push
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 705.90 KiB | 7.84 MiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/RxDnzMC/praktikum-apl.git
 667fabec..c19c9d9 master -> master
```

Gambar 5.1 Pemakaian GIT

1. git add .

Digunakan untuk menambahkan semua perubahan ke dalam staging area agar siap untuk dikomit.

2. git commit -m "Post-Test-5"

Perintah ini berfungsi untuk menyimpan perubahan yang telah ditambahkan ke staging area, dengan pesan sebagai deskripsi perubahan.

3. git push

Digunakan untuk mengunggah kode ke GitHub.