

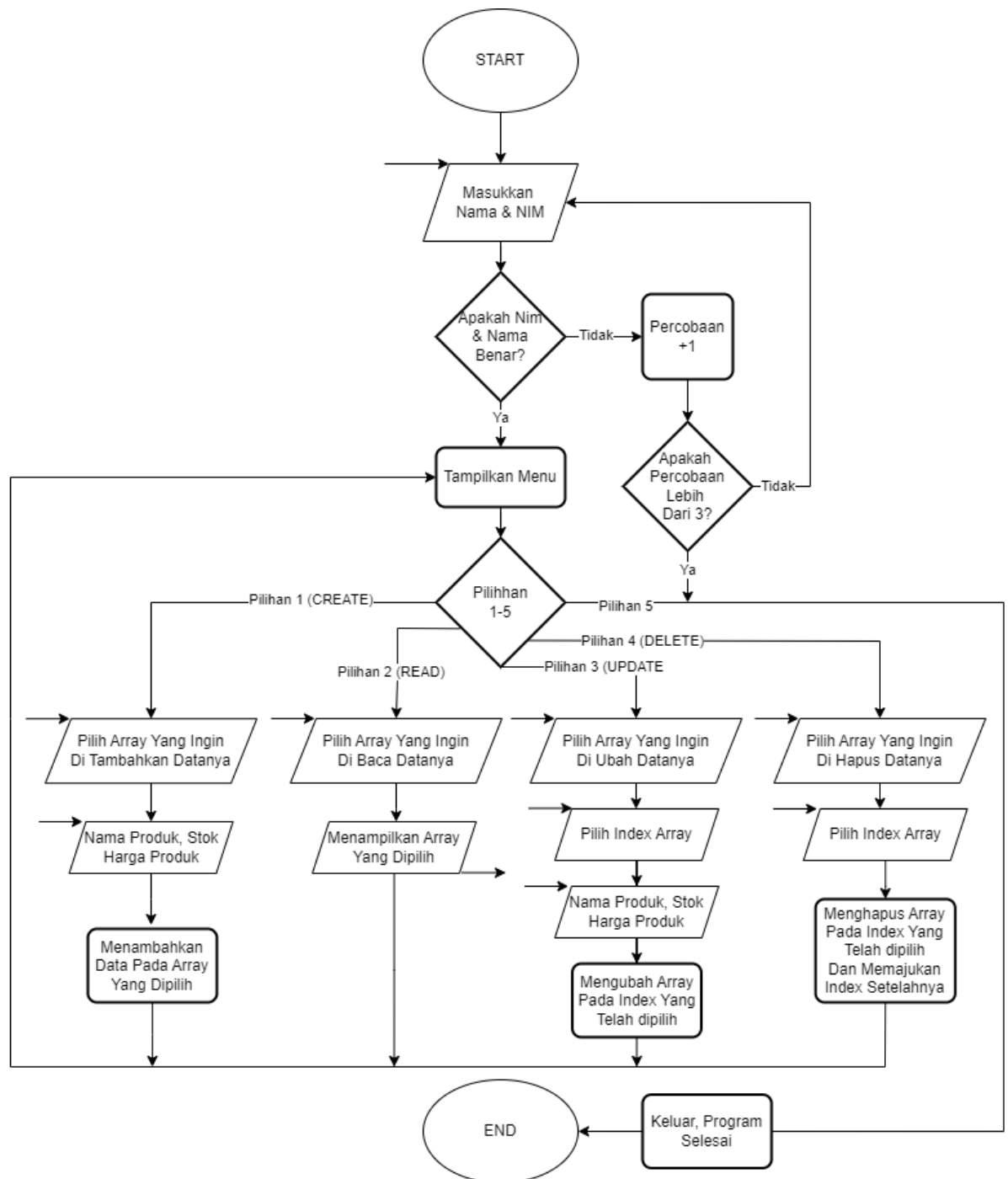
LAPORAN PRAKTIKUM
POSTTEST 2
ALGORITMA PEMROGRAMAN LANJUT



Disusun oleh:
Nama (2409106044)
Kelas (A2 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



Gambar 1.1 Flowchart Program

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.

2.2 Penjelasan Alur & Algoritma

Algoritma:

- Program dimulai dengan meminta pengguna untuk login menggunakan nama dan tiga digit terakhir NIM yang telah ditentukan.
- Jika login berhasil, pengguna diarahkan ke menu utama.
- Jika login gagal, pengguna diberikan tiga kali percobaan, dan jika tiga percobaan gagal, program akan berhenti.
- Menu utama menyediakan lima pilihan:
 1. Menambahkan data barang.
 2. Membaca data barang.
 3. Memperbarui data barang yang sudah ada.
 4. Menghapus data barang.
 5. Keluar dari program.
- Pengguna memilih salah satu opsi yang tersedia.
- Jika pengguna memilih untuk menambahkan, membaca, memperbarui, atau menghapus barang, program akan meminta kategori barang (PlayStation, Xbox, atau Nintendo) sebelum melanjutkan ke program.

- Data barang disimpan dalam array dua dimensi dengan kapasitas maksimum 100 per kategori.
- Program terus berjalan hingga pengguna memilih keluar.

Penjelasan Detail Blok Code/Fungsi Pada Program:

Login:

- Pengguna harus memasukkan nama dan tiga angka terakhir NIM.
- Nama yang digunakan adalah "Rangga" dan tiga angka terakhir NIM adalah "044".
- Pengguna memiliki maksimal tiga kali percobaan untuk login.
- Jika gagal dalam tiga percobaan, program akan berhenti.
- Jika login berhasil, pengguna diarahkan ke menu utama.

Menu Utama:

- Terdiri dari lima opsi utama:
 1. Menambahkan Data Barang
 2. Membaca Data Barang
 3. Memperbarui Data Barang
 4. Menghapus Data Barang
 5. Keluar dari Program

CRUD (Create, Read, Update, Delete):

- Menambahkan Data Barang:
 1. Pengguna memilih kategori (PlayStation, Xbox, atau Nintendo).
 2. Jika kapasitas array belum penuh, pengguna dapat memasukkan nama, stok, dan harga barang.
 3. Data yang dimasukkan disimpan ke dalam array yang sesuai dengan kategori yang dipilih.

- Membaca Data Barang:
 1. Pengguna memilih kategori.
 2. Program menampilkan daftar barang dalam kategori yang dipilih beserta stok dan harga.
 3. Jika tidak ada data dalam kategori tersebut, program memberi notifikasi bahwa data kosong.
- Memperbarui Data Barang:
 1. Pengguna memilih kategori.
 2. Program menampilkan daftar barang dalam kategori yang dipilih.
 3. Pengguna memilih nomor barang yang ingin diperbarui.
 4. Pengguna memasukkan data baru (nama, stok, harga) untuk menggantikan data lama.
- Menghapus Data Barang:
 1. Pengguna memilih kategori.
 2. Program menampilkan daftar barang dalam kategori tersebut.
 3. Pengguna memilih barang yang ingin dihapus.
 4. Barang yang dipilih akan dihapus dari daftar dan data yang tersisa akan digeser ke atas untuk mengisi celah kosong.

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;
    } 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:

```
case 1: // Membuat
    cout << "\nPilih Kategori:\n";
    cout << "1. PlayStation\n2. Xbox\n3. Nintendo\n";
    cout << "Masukkan pilihan: ";
    cin >> kategori;
    if (kategori == 1) {
        cout << "\n";
        cin.ignore();
        if (indexps < MAX_ARRAY) {
            cout << "Masukkan nama item : ";
        }
    }
}
```

```

        getline (cin, playstation[indexps][0]);
        cout << "Masukkan stock item : ";
        getline (cin, playstation[indexps][1]);
        cout << "Masukkan harga item : ";
        getline (cin, playstation[indexps][2]);
        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][0]);
        cout << "Masukkan stock item : ";
        getline (cin, xbox[indexxbox][1]);
        cout << "Masukkan harga item : ";
        getline (cin, xbox[indexxbox][2]);
        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][0]);
        cout << "Masukkan stock item : ";
        getline (cin, nintendo[indexnintendo][1]);
        cout << "Masukkan harga item : ";
        getline (cin, nintendo[indexnintendo][2]);
        indexnintendo++;
    }
    else {
        cout << "Data Array Penuh!!";
    }
}
else {
    cout << "Pilihan tidak valid! Coba lagi." << endl;
}
break;

```

C. Membaca Data Array (READ)

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

Source Code:

```
case 2: // Membaca
    cout << "\nPilih Kategori:\n";
    cout << "1. PlayStation\n2. Xbox\n3. Nintendo\n";
    cout << "Masukkan pilihan: ";
    cin >> 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][0] << " | "
                << playstation[i][1] << " | "
                << playstation[i][2] << " | \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][0] << " | "
                << xbox[i][1] << " | "
                << xbox[i][2] << " | \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][0] << " | "
                << nintendo[i][1] << " | "
                << nintendo[i][2] << " | \n";
        }
        cout << "=====\n";
    } else {
```



```

        cout << "Pilihan tidak valid! Coba lagi.\n";
    }
    break;

```

D. Mengubah Data Array (UPDATE)

Bagian Ini Adalah Bagian Mengubah Data Array Sesuai Index Yang Dipilih Dan Juga Kategori Yang Di Pilih.

Source Code :

```

case 3: // Update
    int pilihanupdate;
    cout << "\nPilih Kategori:\n";
    cout << "1. PlayStation\n2. Xbox\n3. Nintendo\n";
    cout << "Masukkan pilihan: ";
    cin >> 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";
            break;
        }

        for (int i = 0; i < indexps; i++) {
            cout << i + 1 << ". " << playstation[i][0] << " " <<
playstation[i][1] << " " << playstation[i][2] << 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";
            break;
        }
        cout << "Masukkan nama item : ";
        getline (cin, playstation[pilihanupdate][0]);
        cout << "Masukkan stock item : ";
        getline (cin, playstation[pilihanupdate][1]);
        cout << "Masukkan harga item : ";

```

```

        getline (cin, playstation[pilihanupdate][2]);
    }

    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";
            break;
        }

        for (int i = 0; i < indexxbox; i++) {
            cout << i + 1 << ". " << xbox[i][0] << " " << xbox[i][1] <<
" " << xbox[i][2] << endl;
        }

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

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

        cout << "Masukkan nama item : ";
        getline(cin, xbox[pilihanupdate][0]);
        cout << "Masukkan stock item : ";
        getline(cin, xbox[pilihanupdate][1]);
        cout << "Masukkan harga item : ";
        getline(cin, xbox[pilihanupdate][2]);
    }

    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";
            break;
        }

        for (int i = 0; i < indexnintendo; i++) {
            cout << i + 1 << ". " << nintendo[i][0] << " " <<
nintendo[i][1] << " " << nintendo[i][2] << 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";
        break;
    }

    cout << "Masukkan nama item : ";
    getline(cin, nintendo[pilihanupdate][0]);
    cout << "Masukkan stock item : ";
    getline(cin, nintendo[pilihanupdate][1]);
    cout << "Masukkan harga item : ";
    getline(cin, nintendo[pilihanupdate][2]);
}

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

```

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 :

```

case 4: // Menghapus
    int pilihanhapus;
    cout << "\nPilih Kategori:\n";
    cout << "1. PlayStation\n2. Xbox\n3. Nintendo\n";
    cout << "Masukkan pilihan: ";
    cin >> 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";
        break;
    }

    for (int i = 0; i < indexps; i++) {
        cout << i + 1 << ". " << playstation[i][0] << " " <<
playstation[i][1] << " " << playstation[i][2] << endl;
    }

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

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

    for (int i = pilihanhapus - 1; i < indexps - 1; i++) {
        playstation[i][0] = playstation[i + 1][0];
        playstation[i][1] = playstation[i + 1][1];
        playstation[i][2] = playstation[i + 1][2];
    }
    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";
        break;
    }

    for (int i = 0; i < indexxbox; i++) {
        cout << i + 1 << ". " << xbox[i][0] << " " << xbox[i][1] <<
" " << xbox[i][2] << endl;
    }

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

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

    for (int i = pilihanhapus - 1; i < indexxbox - 1; i++) {
        xbox[i][0] = xbox[i + 1][0];
        xbox[i][1] = xbox[i + 1][1];

```

```

        xbox[i][2] = xbox[i + 1][2];
    }
    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";
        break;
    }

    for (int i = 0; i < indexnintendo; i++) {
        cout << i + 1 << ". " << nintendo[i][0] << " " <<
nintendo[i][1] << " " << nintendo[i][2] << endl;
    }

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

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

    for (int i = pilihanhapus - 1; i < indexnintendo - 1; i++) {
        nintendo[i][0] = nintendo[i + 1][0];
        nintendo[i][1] = nintendo[i + 1][1];
        nintendo[i][2] = nintendo[i + 1][2];
    }
    indexnintendo--;
}
else {
    cout << "Pilihan kategori tidak valid!\n";
}
break;

```

4. Uji Coba dan Hasil Output

4.1 Uji Coba

1. Skenario 1

- Login Sembarang
- Login 3x gagal
- Program berhenti

2. Skenario 2

- Login Sesuai NIM dan Nama
- Login Berhasil
- Memilih Menambahkan Data (pilihan 1)
- Memilih Kategori Playstation (pilihan 1)
- Input Nama Produk
- Input Stok Produk
- Input Harga Produk
- Memilih Menampilkan Data (pilihan 2)
- Pilih Kategori Playstation
- Memilih Mengubah Data (pilihan 3)
- Memilih Kategori Playstation (pilihan 1)
- Memilih Index, Menginput 1 (index 0)
- Input Nama Produk
- Input Stok Produk
- Input Harga Produk
- Memilih Menampilkan Data (pilihan 2)
- Pilih Kategori Playstation

- Memilih Menghapus Data (pilihan 4)
- Pilih Kategori Playstation
- Pilih Index
- Memilih Menampilkan Data (pilihan 2)
- Pilih Kategori Playstation
- Memilih Keluar (pilihan 5)

4.2 Hasil Output

Skenario 1:

```
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 Skenario 1 (Gagal)

Skenario 2: Login Berhasil Dan Melakukan CRUD Pada Kategori Playstation

```

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 4.1 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 4.2 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 4.3 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 4.4 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 4.5 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 4.6 Mengecek Yang Sudah Terhapus

5. Git

```
PS C:\Users\USER\Documents\.Tugas Kuliah\codingan\C++\pratikum-apl> git add .
PS C:\Users\USER\Documents\.Tugas Kuliah\codingan\C++\pratikum-apl> git commit -m "Post-Test-2"
[master 70e70fa] Post-Test-2
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 post-test/post-test-apl-2/2409106044-RanggaAdityaRahman-PT-2.pdf
PS C:\Users\USER\Documents\.Tugas Kuliah\codingan\C++\pratikum-apl> git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 524.27 KiB | 30.84 MiB/s, done.
Total 6 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/RxDnzMC/praktikum-apl.git
 a8925db..70e70fa master -> master
PS C:\Users\USER\Documents\.Tugas Kuliah\codingan\C++\pratikum-apl> █
```

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-2"

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.