**BAB I**

**LANDASAN TEORI**

* 1. **Linked List**

Linked List adalah struktur data linier dari elemen data, yang terdiri dari urutan record data dimana setiap record memiliki field yang menyimpan alamat/referensi dari record selanjutnya (dalam urutan). Linked list dibentuk secara dinamis, karena memori yang digunakan tidak berurutan. Biasanya didalam suatu linked list, terdapat istilah head dan tail (dapat diganti dengan variabel lain), head merupakan elemen yang berada pada posisi pertama, sedangkan tail elemen yang berada pada akhir suatu linked list. Perbedaan mencolok antara linked list dan array ada pada penggunaan memorinya. Jika pada array memori yang digunakan berurutan dan jumlah memori tidak bisa ditambah, sedangkan pada linked list memori tidak berurutan dan dapat ditambah. Adapun jenis – jenis linked list sebagai berikut.

* + 1. **Single Linked List**

Single linked list atau biasa disebut linked list terdiri dari elemen-elemen individu, dimana masing-masing dihubungkan dengan pointer tunggal. Masing-masing elemen terdiri dari dua bagian, yaitu sebuah data dan sebuah pointer yang disebut dengan pointer next.

Next pointer

20

13

4

25

Tail

Head

* + 1. **Double Linked List**

Double Linked List merupakan suatu linked list yang memiliki dua variabel pointer yaitu pointer yang menunjuk ke node selanjutnya dan pointer yang menunjuk ke node sebelumnya. Setiap head dan tailnya juga menunjuk ke NULL.

Previous

Next

20

13

4

25

Tail

Head

* + 1. **Circular Linked List**

Circular Linked List merupakan suatu linked list dimana tail (node terakhir) menunjuk ke head (node pertama). Jadi tidak ada pointer yang menunjuk NULL. Ada 2 jenis Circular Linked List, yaitu :

* **Circular Single Linked List**

Next pointer

20

13

4

25

Tail

Headddd

* **Circular Double Linked List**

Tail

Previous

Next

20

13

4

25

Head

* 1. **Input dan Output File**

Pada dasarnya, operasi file terbagi 3 jenis. **Read (R), Write(W), dan Append(A)**. Read artinya membaca isi file, write menulis data ke file dan append menambahkan data ke baris terakhir dalam file.

**Struktur Dasar**

int main(){

FILE \*in=fopen(namafile,mode);

fclose(in);

getchar();

return 0;

}

* **FILE \*in=fopen(namafile, mode);** pertama membuka file terlebih dahulu dengan perintah fopen. Fungsi fopen diikuti dengan 2 variabel diantaranya namafile dan mode.
* Setelah dibuka, file tersebut wajib ditutup dengan perintah fclose(nama variabel). Karena pada awal membuka file dengan variabel pointer “in”, maka diakhir program juga wajib menutup file dengan fopen(in);

| **Parameter Mode** | **Arti** |
| --- | --- |
| r | Membuka file sesuai namafilenya untuk dibaca, jika file tidak ada maka akan bernilai NULL. |
| w | Membuka file untuk ditulis, jika sebelumnya sudah ada nama file yang sama, maka akan dilakukan overwrite/menimpa file dengan yang baru. |
| a | Membuka file sesuai namafilenya untuk ditambahkan datanya dipaling bawah baris terakhir. Jika file tidak ada, maka akan membuat file baru secara otomatis. |
| r+ | Sama dengan “r” tetapi selain file dapat dibaca juga dapat di tulis |
| w+ | Sama dengan “w” tetapi selain file dapat ditulis juga dapat dibaca |
| a+ | Sama dengan ”a” tetapi selain file dapat ditulis file juga dapat dibaca |

* + 1. **Read File**

Untuk membaca isi file lalu menyimpannya ke dalam variabel, dapat menggunakan fscanf(namavariabelfile, format);

Contoh  Isi file test.txt (letakkan di folder yang sama dengan file .cpp/.c) :

int main(){

char nama[100];

int umur;

FILE \*in=fopen("test.txt","r");

while(!feof(in)){

fscanf(in,"%[^#]#%d\n", &nama, &umur);fflush(stdin);

// %[^#] artinya menyimpan bagian dari string dalam file sampai tanda #.

// tidak menggunnakan %s karena nama mengandung spasi

printf("%s %d\n", nama, umur);

}

fclose(in);

getchar();

return 0;

}

Saat read file, tidak diperlukan penambahan fclose terhadap file yang tidak ada. Karena akan mengakibatkan error.

* + 1. **Write File**

Write file artinya membuka file dalam mode write (siap untuk ditulis). Berikut merupakan contoh program write file.

int main(){

char nama[100];

int umur;

printf("Masukkan nama : "); scanf("%[^\n]", &nama); fflush(stdin);

printf("Masukkan umur : "); scanf("%d", &umur); fflush(stdin);

FILE \*out=fopen("test.txt","w");

fprintf(out,"%s#%d\n",nama, umur);

fclose(out);

printf("Sukses menambah data.");

getchar();

return 0;

}

Script diatas jika dijakankan maka akan membuat file baru jika sudah ada datanya **(overwrite)**. Lalu, hasil inputan nama dan umur akan disimpan dalam file test.txt dengan format NAMA#UMUR.

* + 1. **Append File**

Append artinya menambahkan data pada file baris terakhir. Jika belum ada data/filenya, maka append akan membuatkan file baru. Contoh :

int main(){

char nama[100];

int umur;

printf("Masukkan nama : "); scanf("%[^\n]", &nama); fflush(stdi);

printf("Masukkan umur : "); scanf("%d", &umur); fflush(stdin);

FILE \*out=fopen("test.txt","a");

fprintf(out,"%s#%d\n",nama, umur);

fclose(out);

printf("Sukses menambah data.");

getchar();

return 0;

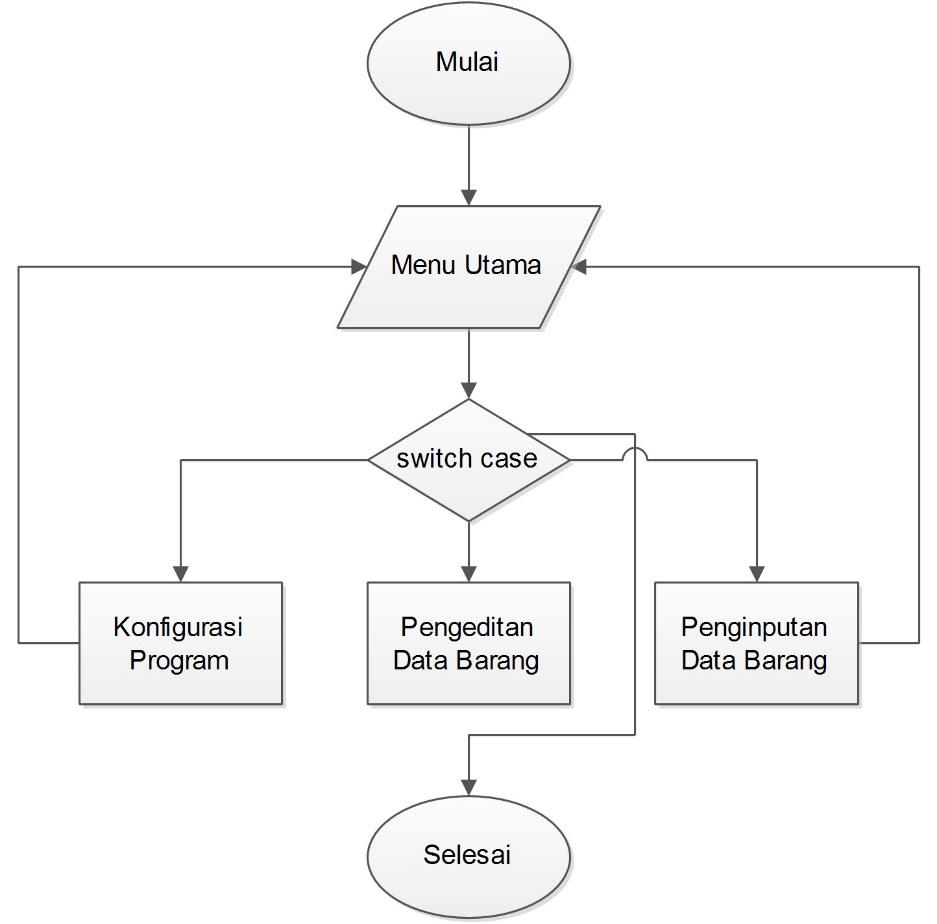
}

**BAB II**

**PEMBAHASAN**

**2.1 Program Manajemen Sederhana**

**2.1.1 Bagan-Alur (*Flowchart*)**

****

Gambar 2.1: Bagan-alur (*flowchart*) program Manajemen Data Sederhana.

**2.1.2 Algoritma Program**

1. Saat program dijalankan akan muncul halaman judul program.
2. Lalu akan menampilkan halaman menu program yang terdiri dari konfigurasi program, pengeditan data barang, penginputan data barang, dan keluar. User akan diminta untuk menginput pilihan.
3. Pada halaman menu menggunakan *switch case,* saat inputan user == 1 , program akan menuju case 1, dst.
4. Pada case 1, terdapat menu terdapat menu lihat informasi toko, ubah, simpan pencapaian terkini, dan kembali. Lalu user akan diminta kembali memasukan inputan.

* Saat inputan user == 1, user akan diminta menginput data informasi toko,diantaranya: nama toko, alamat, dll.
* Saat inputan user == 2, akan menampilkan pengeditan data informasi toko yang sudah diinput.
* Saat input user ==3, akan menyimpan hasil data yang masuk

Semua inputan akan disimpan pada file yang akan dibuat oleh program. Dengan menggunakan metode linked list.

1. Pada case 2, terdapat pilihan edit barang dan hapus barang. User akan diminta menginput kembali.

* Saat pilihan user == 1, akan menampilkan menu edit data barang masuk dan keluar
* Saat pilihan user == 2, akan menghapus data barang yang telah diinput user

1. Pada case 3, merupakan menu penginputan data barang, yang berisi manajemen data barang dan lihat semua transaksi.

* Saat pilihan user == 1, maka akan meminta inputan barang masuk.
* Saat pilihan user == 2, akan melihat semua transaksi yang telah diinputkan user

Inputan data barang menggunakan metode linked list dalam pengolahan memori.

1. Program Selesai

**2.1.3 Program**

* 1. *Source Code*

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

#include <ctype.h>

#include <string.h>

int akun = 0, hitung\_masuk = 0, hitung\_keluar = 0, stok;

char yakin, pilih\_exit;

struct simpul\_info\_toko

{

char nama\_toko[20];

char alamat[50];

char kota[20];

char npwp[20];

char nama\_barang[20];

char satuan\_grosir[10];

int harga\_beli;

int harga\_jual;

int kapasitas\_max;

char mata\_uang[3];

}info\_toko;

struct password

{

char password\_admin[8];

char password\_supervisor[8];

char pil\_ganti\_password;

char admin\_sementara[8];

char supervisor\_sementara[8];

}head\_password;

struct simpul\_masuk

{

int ID;

char nama[20];

int jml;

char pil\_simpan[1];

struct simpul \*next;

}\*head\_data\_barang\_masuk = NULL;

struct simpul\_masuk \*temp\_data\_barang\_masuk = NULL;

struct simpul\_masuk \*temp\_data\_barang\_keluar = NULL;

struct simpul\_masuk \*search\_data\_barang\_masuk = NULL;

struct simpul\_masuk \*search\_data\_barang\_keluar = NULL;

struct simpul\_keluar

{

int ID;

char nama[20];

int jml;

char pil\_simpan[1];

struct simpul \*next;

}\*head\_data\_barang\_keluar = NULL;

struct simpul\_supervisor

{

int nomor\_data\_diedit;

int pil\_edit\_data;

struct simpul \*next;

}\*supervisor = NULL;

void selamat\_datang();

void header();

void list\_akun();

int pilihan\_akun();

void akun\_administrator();

void akun\_supervisor();

void akun\_user();

void ubah\_info\_toko();

void lihat\_info\_toko();

int main()

{

selamat\_datang();

while (akun == 0)

{

system("cls");

header();

list\_akun();

akun = pilihan\_akun();

switch(akun)

{

case 1:

system("cls");

header();

akun\_administrator();

akun = 0;

break;

case 2:

system("cls");

header();

akun\_supervisor();

akun = 0;

break;

case 3:

system("cls");

header();

akun\_user();

akun = 0;

break;

case 4:

exit\_prompt();

break;

default:

printf("\nMaaf, inputan salah!\n");

system("pause");

akun = 0;

break;

}

}

return 0;

}

void selamat\_datang()

{

system("cls");

printf("Selamat datang di program ini!\n\n");

system("pause");

}

void header()

{

printf("HEADER\n\n");

}

void list\_akun()

{

printf("Pilihan:\n");

printf("1. Konfigurasi Program\n");

printf("2. Pengeditan Data Barang\n");

printf("3. Penginputan Data Barang\n");

printf("4. Keluar\n\n");

printf("Pilih nomor : ");

}

int pilihan\_akun()

{

int pil\_akun;

fflush(stdin);

scanf("%d", &pil\_akun);

return pil\_akun;

}

void akun\_administrator()

{

system("cls");

header();

int pil\_admin;

printf("Selamat datang di akun Administrator!\n\n");

printf("Pilihan:\n");

printf("1. Lihat informasi toko\n");

printf("2. Ubah informasi toko\n");

printf("3. Ubah kata kuncir\n");

printf("4. Keluar dari akun\n\n");

printf("Pilih nomor : ");

scanf("%d", &pil\_admin);

switch(pil\_admin)

{

case 1:

lihat\_info\_toko();

break;

case 2:

ubah\_info\_toko();

break;

case 3:

ubah\_kata\_kunci();

break;

case 4:

return 0;

break;

default:

printf("Maaf, inputan salah.");

break;

}

}

void akun\_supervisor()

{

int pil\_supervisor;

do

{

system("cls");

header();

printf("Selamat datang di akun Supervisor!\n\n");

scanf("%d", &pil\_supervisor);

switch(pil\_supervisor)

{

case 1:

printf("1");

break;

case 2:

printf("2");

break;

case 3:

printf("3");

break;

}

}

while(pil\_supervisor != 4);

system("pause");

akun = 0;

return list\_akun();

}

void akun\_user()

{

int pilihan\_akun\_user;

system("cls");

header();

printf("Selamat datang di akun User!\n\nPilihan aksi:\n1. Manajemen data barang\n2. Lihat semua transaksi\n3. Keluar dari akun\n\nPilih nomor : ");

scanf("%d", &pilihan\_akun\_user);

switch(pilihan\_akun\_user)

{

case 1:

system("cls");

header();

int pil\_manajemen;

printf("Pilihan:\n1. Barang masuk (pembelian)\n2. Barang keluar (penjualan)\n3. Kembali");

scanf("%d", &pil\_manajemen);

switch(pil\_manajemen)

{

case 1:

system("cls");

header();

input\_barang(1);

break;

case 2:

system("cls");

header();

input\_barang(2);

break;

case 3:

system("pause");

break;

}

break;

case 2:

break;

case 3:

break;

}

system("pause");

}

exit\_prompt()

{

do

{

system("cls");

printf("Yakin ingin keluar dari program?\nTekan 'Y' untuk melanjutkan : ");

fflush(stdin);

scanf("%c", &pilih\_exit);

toupper(pilih\_exit);

}

while (pilih\_exit != 'Y');

if(pilih\_exit == 'Y')

exit(1);

}

void ubah\_info\_toko()

{

FILE \*fptr;

fptr = fopen("info\_toko.txt", "w+");

printf("Nama toko : ");

fflush(stdin);

gets(info\_toko.nama\_toko);

fprintf(fptr, "Nama toko: %s\n", info\_toko.nama\_toko);

printf("Alamat : ");

fflush(stdin);

gets(info\_toko.alamat);

fprintf(fptr, "Alamat: %s\n", info\_toko.alamat);

printf("Kota : ");

fflush(stdin);

gets(info\_toko.kota);

fprintf(fptr, "Kota: %s\n", info\_toko.kota);

printf("NPWP : ");

fflush(stdin);

gets(info\_toko.npwp);

fprintf(fptr, "NPWP: %s\n", info\_toko.npwp);

printf("Nama barang : ");

fflush(stdin);

gets(info\_toko.nama\_barang);

fprintf(fptr, "Nama\_barang: %s\n", info\_toko.nama\_barang);

printf("Satuan grosir : ");

fflush(stdin);

gets(info\_toko.satuan\_grosir);

fprintf(fptr, "Satuan\_grosir: %s\n", info\_toko.satuan\_grosir);

printf("Harga beli (per %s) : ", info\_toko.satuan\_grosir);

scanf("%d", &info\_toko.harga\_beli);

fprintf(fptr, "Harga\_beli: %d\n", info\_toko.harga\_beli);

printf("Harga jual (per %s) : ", info\_toko.satuan\_grosir);

scanf("%d", &info\_toko.harga\_jual);

fprintf(fptr, "Harga\_jual: %d\n", info\_toko.harga\_jual);

printf("Kapasitas stok barang maksimum (%s) : ", info\_toko.satuan\_grosir);

scanf("%d", &info\_toko.kapasitas\_max);

fprintf(fptr, "Kapasitas\_stok\_barang\_maksimum: %d\n", info\_toko.kapasitas\_max);

printf("Mata uang : ");

fflush(stdin);

gets(info\_toko.mata\_uang);

fprintf(fptr, "Mata\_uang: %s\n", info\_toko.mata\_uang);

fclose(fptr);

printf("\nInformasi toko berhasil diubah!");

system("pause");

}

void lihat\_info\_toko()

{

FILE \*fptr;

char ch;

fptr = fopen("info\_toko.txt", "r");

if(fptr == NULL)

{

printf("Maaf, informasi toko tidak ditemukan!\n");

system("pause");

}

else

{

printf("Informasi toko :\n\n");

ch = fgetc(fptr);

while(ch != EOF)

{

printf("%c", ch);

ch = fgetc(fptr);

}

system("pause");

}

fclose(fptr);

}

void ubah\_kata\_kunci()

{

FILE \*fd;

FILE \*fp;

system("cls");

header();

printf("Pilihan:\n1. Akun Administrator\n2. Akun Supervisor\n3. Kembali\n\nPilh nomor akun yang ingin kata kuncinya diganti : ");

scanf("%c", &head\_password.pil\_ganti\_password);

switch(head\_password.pil\_ganti\_password)

{

case 1:

fd = fopen("password\_admin.txt", "r");

if(NULL == fd)

{

printf("Error! Tidak ada password yang tersimpan!");

system("pause");

}

else

{

fgets(head\_password.password\_admin, 8, (FILE\*)fd);

system("cls");

printf("Masukkan password lama : ");

fflush(stdin);

gets(head\_password.admin\_sementara);

if(head\_password.admin\_sementara == head\_password.password\_admin)

{

printf("Masukkan password baru : ");

fflush(stdin);

gets(head\_password.password\_admin);

fclose(fd);

fp = fopen("password\_admin.txt", "w");

fputs(head\_password.password\_admin, (FILE\*)fd);

fclose(fp);

printf("Password akun Administrator berhasil diubah!\n");

system("pause");

}

else

{

printf("Maaf, password lama yang dimasukkan salah!");

system("pause");

fclose(fd);

}

}

break;

case 2:

fd = fopen("password\_supervisor.txt", "r");

if(NULL == fd)

{

printf("Error! Tidak ada password yang tersimpan!");

system("pause");

}

else

{

fgets(head\_password.password\_supervisor, 8, (FILE\*)fd);

system("cls");

printf("Masukkan password lama : ");

fflush(stdin);

gets(head\_password.supervisor\_sementara);

if(head\_password.supervisor\_sementara == head\_password.password\_supervisor)

{

printf("Masukkan password baru : ");

fflush(stdin);

gets(head\_password.password\_supervisor);

fclose(fd);

fp = fopen("password\_admin.txt", "w");

fputs(head\_password.password\_supervisor, (FILE\*)fd);

fclose(fp);

printf("Password akun Supervisor berhasil diubah!\n");

system("pause");

}

else

{

printf("Maaf, password lama yang dimasukkan salah!");

system("pause");

fclose(fd);

}

}

break;

}

}

int masukkan\_password(int pil\_akun)

{

FILE \*fd;

FILE \*fp;

switch (pil\_akun)

{

case 1:

fd = fopen("password\_admin.txt", "r");

if(NULL == fd)

{

printf("Tidak ada kata kunci yang tersimpan.\nMasukkan kata kunci baru (maks. 8 karakter : ");

fflush(stdin);

gets(head\_password.password\_admin);

fclose(fd);

fp = fopen("password\_admin.txt", "w");

fputs(head\_password.password\_admin, (FILE\*)fp);

printf("Kata kunci akun Administrator berhasil disimpan!\n");

system("pause");

}

else if(NULL != fd)

{

system("cls");

printf("Untuk melanjutkan, masukkan kata kunci : ");

fflush(stdin);

gets(head\_password.admin\_sementara);

fflush(stdin);

fd = fopen("password\_admin.txt", "r");

fgets(head\_password.password\_admin, 8, (FILE\*)fd);

if(head\_password.admin\_sementara == head\_password.password\_admin)

{

fclose(fd);

return 9;

}

else

{

fclose(fd);

return 8;

}

}

break;

case 2:

fd = fopen("password\_supervisor.txt", "r");

if(NULL == fd)

{

printf("Tidak ada kata kunci yang tersimpan.\nMasukkan kata kunci baru (maks. 8 karakter : ");

fflush(stdin);

gets(head\_password.password\_supervisor);

fclose(fd);

fp = fopen("password\_supervisor.txt", "w");

fputs(head\_password.password\_supervisor, (FILE\*)fp);

printf("Kata kunci akun Supervisor berhasil disimpan!");

system("pause");

}

else

{

system("cls");

printf("Untuk melanjutkan, masukkan kata kunci : ");

fflush(stdin);

gets(head\_password.supervisor\_sementara);

fgets(head\_password.password\_supervisor, 8, (FILE\*)fd);

if(head\_password.supervisor\_sementara == head\_password.password\_supervisor)

{

fclose(fd);

return 0;

}

else

{

fclose(fd);

return 1;

}

}

break;

}

}

int apakah\_linkedlist\_databarang\_ada(int jenis\_data)

{

switch(jenis\_data)

{

case 1:

if(head\_data\_barang\_masuk == NULL)

return 0;

else

return 1;

break;

case 2:

if(head\_data\_barang\_keluar == NULL)

return 0;

else

return 1;

break;

}

}

void edit\_data\_barang()

{

do

{

system("cls");

header();

printf("Pilihan:\n1. Data barang masuk\n2. Data barang keluar\n3. Kembali\n\nPilih nomor : ");

scanf("%d", &supervisor -> pil\_edit\_data);

switch(supervisor -> pil\_edit\_data)

{

case 1:

system("cls");

header();

switch(apakah\_linkedlist\_databarang\_ada(1))

{

case 0:

printf("Data barang masih kosong!");

break;

case 1:

tampilkan\_data\_barang(1);

printf("\nPilih nomor data yang ingin diedit : ");

scanf("%d", &supervisor -> nomor\_data\_diedit);

if(supervisor -> nomor\_data\_diedit <= 0 || supervisor -> nomor\_data\_diedit > hitung\_masuk)

printf("Maaf, inputan salah!");

else

{

edit\_barang\_spesifik(1, supervisor -> nomor\_data\_diedit);

printf("Edit data barang masuk nomor %d berhasil!", supervisor -> nomor\_data\_diedit);

}

system("pause");

break;

}

case 2:

system("cls");

header();

switch(apakah\_linkedlist\_databarang\_ada(1))

{

case 0:

printf("Data barang masih kosong!");

break;

case 1:

tampilkan\_data\_barang(2);

printf("\nPilih nomor data yang ingin diedit : ");

scanf("%d", &supervisor -> nomor\_data\_diedit);

if(supervisor -> nomor\_data\_diedit <= 0)

printf("Maaf, inputan salah!");

else

{

edit\_barang\_spesifik(2, supervisor -> nomor\_data\_diedit);

printf("Edit data barang masuk nomor %d berhasil!", supervisor -> nomor\_data\_diedit);

}

system("pause");

break;

}

case 3:

return 1;

break;

}

}

while(1);

}

int input\_barang(int jenis\_data)

{

struct simpul\_masuk \*simpul\_masuk\_baru;

struct simpul\_keluar \*simpul\_keluar\_baru;

switch(jenis\_data)

{

case 1:

simpul\_masuk\_baru = (struct simpul\_masuk \*)malloc(sizeof(struct simpul\_masuk));

do

{

system("cls");

header();

hitung\_masuk++;

simpul\_masuk\_baru -> ID = hitung\_masuk;

printf("Masukkan nama supplier : ");

fflush(stdin);

gets(simpul\_masuk\_baru -> nama);

printf("Masukkan jumlah barang masuk (%s) : ", info\_toko.satuan\_grosir);

scanf("%d", &simpul\_masuk\_baru -> jml);

if(head\_data\_barang\_masuk == NULL)

head\_data\_barang\_masuk = simpul\_masuk\_baru;

else

{

temp\_data\_barang\_masuk = head\_data\_barang\_masuk;

while(temp\_data\_barang\_masuk->next != NULL)

temp\_data\_barang\_masuk = temp\_data\_barang\_masuk->next;

temp\_data\_barang\_masuk->next = simpul\_masuk\_baru;

}

printf("\n\nData barang masuk berhasil disimpan!\nIngin memasukkan yang lain? Tekan 'Y' : ");

fflush(stdin);

scanf("%c", &simpul\_masuk\_baru -> pil\_simpan);

toupper(simpul\_masuk\_baru -> pil\_simpan);

}

while(simpul\_masuk\_baru -> pil\_simpan =='Y');

break;

case 2:

simpul\_keluar\_baru = (struct simpul\_keluar \*)malloc(sizeof(struct simpul\_keluar));

do

{

system("cls");

header();

hitung\_keluar++;

simpul\_keluar\_baru -> ID = hitung\_keluar;

printf("Masukkan nama pembeli : ");

fflush(stdin);

gets(simpul\_keluar\_baru -> nama);

printf("Masukkan jumlah barang keluar (%s) : ", info\_toko.satuan\_grosir);

scanf("%d", &simpul\_keluar\_baru -> jml);

if(head\_data\_barang\_keluar == NULL)

head\_data\_barang\_keluar = simpul\_keluar\_baru;

else

{

temp\_data\_barang\_keluar = head\_data\_barang\_keluar;

while(temp\_data\_barang\_keluar->next != NULL)

temp\_data\_barang\_keluar = temp\_data\_barang\_keluar->next;

temp\_data\_barang\_keluar->next = simpul\_keluar\_baru;

}

printf("\n\nData barang keluar berhasil disimpan!\nIngin memasukkan yang lain? Tekan 'Y' : ");

fflush(stdin);

scanf("%c", &simpul\_keluar\_baru -> pil\_simpan);

toupper(simpul\_keluar\_baru -> pil\_simpan);

}

while(simpul\_keluar\_baru -> pil\_simpan =='Y');

break;

}

}

int tampilkan\_data\_barang (int jenis\_data)

{

header\_tabel();

cetak\_garis(49);

switch(jenis\_data)

{

case 1:

temp\_data\_barang\_masuk = head\_data\_barang\_masuk;

while (temp\_data\_barang\_masuk != NULL)

{

printf(" %d\t| %s\t| %d\t| %d\n", temp\_data\_barang\_masuk -> ID, temp\_data\_barang\_masuk -> nama, temp\_data\_barang\_masuk -> jml, temp\_data\_barang\_masuk -> jml \* info\_toko.harga\_beli);

temp\_data\_barang\_masuk = temp\_data\_barang\_masuk -> next;

}

printf("\n");

break;

case 2:

temp\_data\_barang\_keluar = head\_data\_barang\_keluar;

while (temp\_data\_barang\_keluar != NULL)

{

printf(" %d\t| %s\t| %d\t| %d\n", temp\_data\_barang\_keluar -> ID, temp\_data\_barang\_keluar -> nama, temp\_data\_barang\_keluar -> jml, temp\_data\_barang\_keluar -> jml \* info\_toko.harga\_jual);

temp\_data\_barang\_keluar = temp\_data\_barang\_keluar -> next;

}

printf("\n");

break;

}

cetak\_garis(49);

}

int edit\_barang\_spesifik(int jenis\_data, int nomor)

{

switch(jenis\_data)

{

case 1:

search\_data\_barang\_masuk = head\_data\_barang\_masuk;

if(search\_data\_barang\_masuk -> ID == nomor)

{

printf("Masukkan nama supplier baru : ");

fflush(stdin);

gets(search\_data\_barang\_masuk -> nama);

printf("Masukkan jumlah barang masuk baru (%s) : ", info\_toko.satuan\_grosir);

scanf("%d", &search\_data\_barang\_masuk -> jml);

}

else

{

while(search\_data\_barang\_masuk -> next != NULL)

{

search\_data\_barang\_masuk = search\_data\_barang\_masuk -> next;

if(search\_data\_barang\_masuk -> ID == nomor)

{

search\_data\_barang\_masuk = search\_data\_barang\_masuk -> next;

printf("Masukkan nama supplier baru : ");

fflush(stdin);

gets(search\_data\_barang\_masuk -> nama);

printf("Masukkan jumlah barang masuk baru (%s) : ", info\_toko.satuan\_grosir);

scanf("%d", &search\_data\_barang\_masuk -> jml);

}

else

search\_data\_barang\_masuk = search\_data\_barang\_masuk -> next;

}

printf("Data barang masuk bernomor %d tidak ditemukan!", nomor);

system("cls");

}

break;

case 2:

search\_data\_barang\_keluar = head\_data\_barang\_keluar;

if(search\_data\_barang\_keluar -> ID == nomor)

{

printf("Masukkan nama pembeli baru : ");

fflush(stdin);

gets(search\_data\_barang\_keluar -> nama);

printf("Masukkan jumlah barang keluar baru (%s) : ", info\_toko.satuan\_grosir);

scanf("%d", &search\_data\_barang\_keluar -> jml);

}

else

{

while(search\_data\_barang\_keluar -> next != NULL)

{

search\_data\_barang\_keluar = search\_data\_barang\_keluar -> next;

if(search\_data\_barang\_keluar -> ID == nomor)

{

search\_data\_barang\_keluar = search\_data\_barang\_keluar -> next;

printf("Masukkan nama pembeli baru : ");

fflush(stdin);

gets(search\_data\_barang\_keluar -> nama);

printf("Masukkan jumlah barang keluar baru (%s) : ", info\_toko.satuan\_grosir);

scanf("%d", &search\_data\_barang\_keluar -> jml);

}

else

search\_data\_barang\_keluar = search\_data\_barang\_keluar -> next;

}

printf("Data barang keluar bernomor %d tidak ditemukan!", nomor);

system("cls");

}

break;

}

}

void cetak\_garis(int panj\_garis)

{

int i;

for(i=1;i<=panj\_garis;i++)

printf("-");

printf("\n");

}

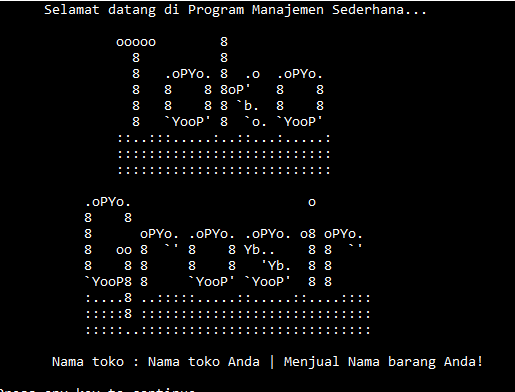
void header\_tabel()

{

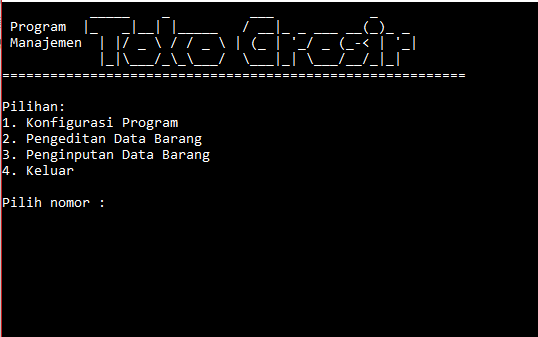
printf(" No\t| Nama\t\t| Jml\t| Harga\n");

}

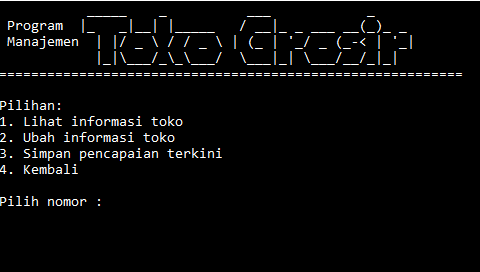
* 1. ***Screenshot* Saat Program Dijalankan**



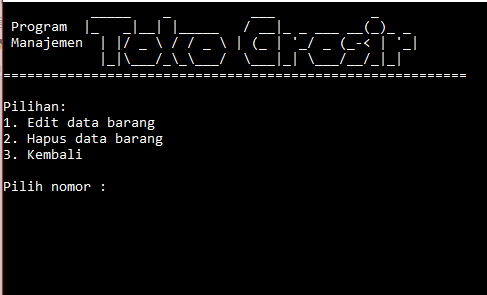
Gambar 2.2: *Screenshot* tampilan nama program



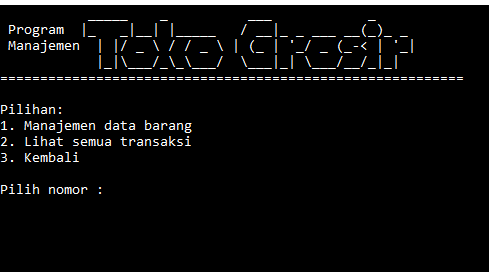
Gambar 2.3: *Screenshot* tampilan menu



Gambar 2.4: *Screenshot* tampilan menu konfigurasi program



Gambar 2.5: *Screenshot* tampilan menu pengeditan data



Gambar 2.6: *Screenshot* menu penginputan data

**DAFTAR PUSTAKA**

Gookin, D. 2014. *Beginning Programming with C for Dummies*. New Jersey. John Wiley & Sons, Inc.

Munir, R. 2016. *Algoritma dan Pemrograman dalam Bahasa Pascal, C, dan C++.* Edisi Keenam*.* Jakarta. Informatika.