LAPORAN PRAKTIKUM

**POSTTEST 4**

**ALGORITMA PEMROGRAMAN LANJUT**

****

**Disusun oleh:**

**Dimas Elang Satria (2409106027)**

**Kelas (A2 ‘24)**

**PROGRAM STUDI INFORMATIKA**

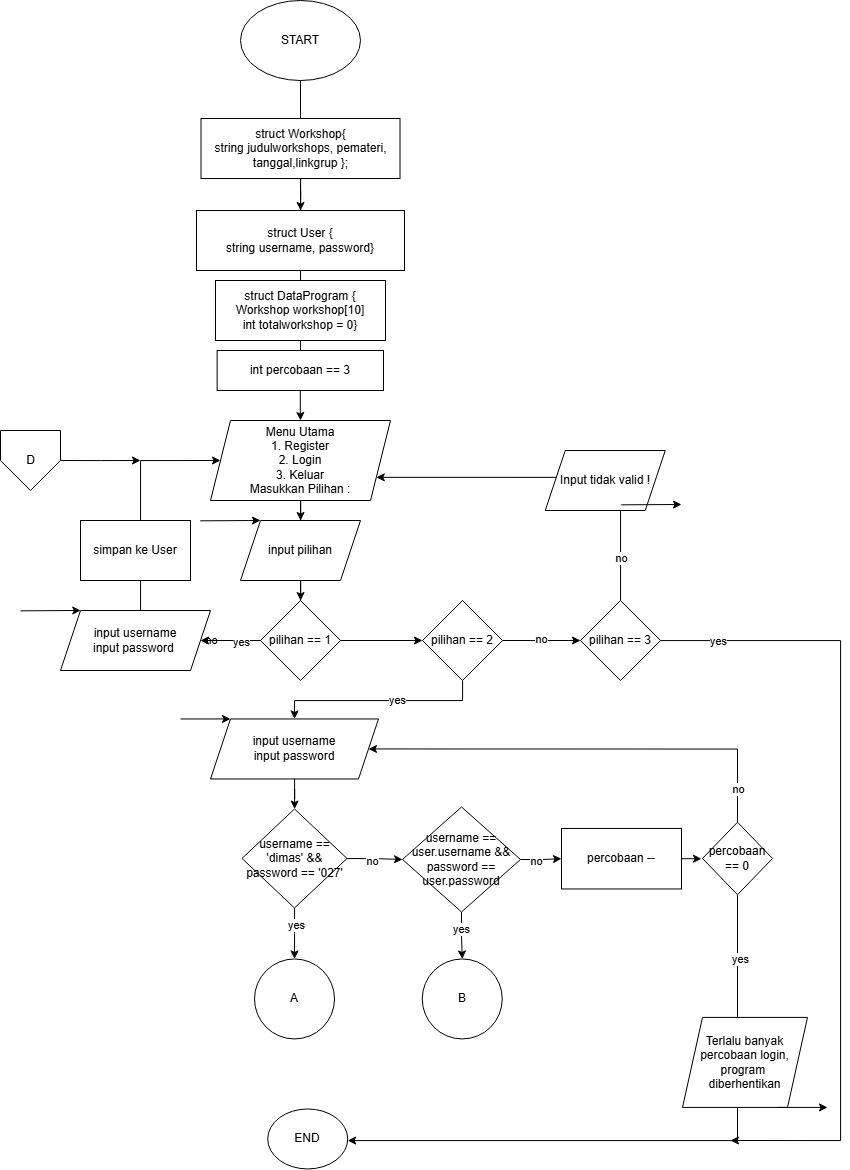
**UNIVERSITAS MULAWARMAN**

**SAMARINDA**

**2025**

# 

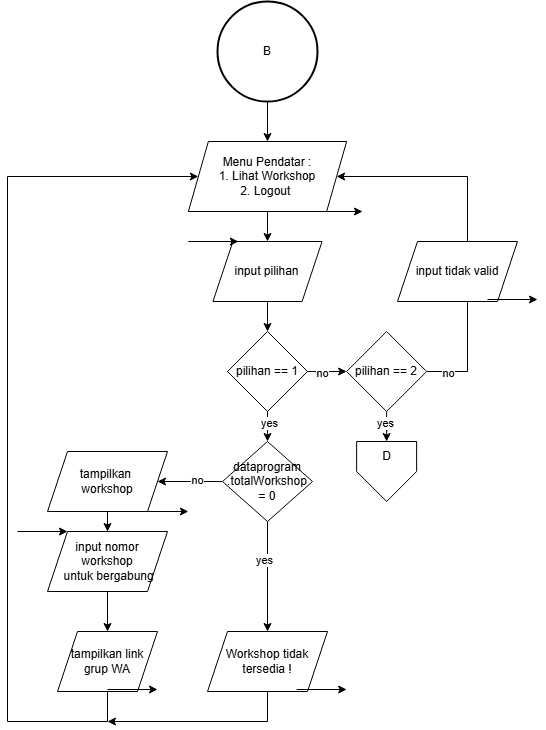
# 1. Flowchart

**

Gambar 1.1 main

# 

Gambar 1.2 login\_admin



Gambar 1.3 login\_pendaftar

# 2. Analisis Program

Program ini bertujuan untuk memudahkan individu dalam mencari workshop, yang bermanfaat. Disini kita dapat mencari banyak macam workshop-workshop dalam satu platform, sehingga dapat meningkatkan efisiensi pencarian. Program ini juga menerapkan multi user dimana admin dapat mengontrol seluruh program (menambah workshop, melihat workshop, merubah workshop, dan menghapus workshop) sedangkan nonadmin bisa melihat-lihat workshop tersedia dan join link grup whatsapp, untuk bisa bergabung di workshop.

Program ini merupakan lanjutan dari postest 2 (membuat CRUD array 1 dimensi).

# 3. Source Code

### **A. Menu Awal**

Fitur ini digunakan untuk menampilkan menu utama dari program yang terdiri atas 3 pilihan.

**Source Code :**

|  |
| --- |
| while (true) {  cout << "\n|| ===== MENU AWAL ===== ||\n";  cout << "1. Register\n2. Login\n3. Keluar\nPilih: ";  cin >> pilihan;  cin.ignore(); if (pilihan == 1) {*//kode register* } else if (pilihan == 2) {*//kode login* } else if (pilihan == 3) {*//kode keluar* } else { cout << "Pilihan tidak valid!\n";  } |

**B. Register**

Fitur ini digunakan untuk membuat user baru hanya sebagai pendaftar sehingga bisa menggunakan program

**Source Code :**

|  |
| --- |
| if (pilihan == 1) {  cout << "\n|| ===== REGISTER ===== ||\n";  cout << "Masukkan username baru : ";  getline(cin, user.username);  cout << "Masukkan Password baru : ";  getline(cin, user.password);  } |

**C. Login**

Fitur ini digunakan untuk mengautentikasi user terdaftar yang akan masuk program apakah admin atau pendaftar dengan memasukkan username dan password yang telah ditentukan.

**Source Code :**

|  |
| --- |
| else if (pilihan == 2) {  string inputusn, inputpw;  cout << "\n|| ===== LOGIN ===== ||\n";  cout << "Masukkan username : ";  getline(cin, inputusn);  cout << "Masukkan Password : ";  getline(cin, inputpw);  if (inputusn == "dimas" && inputpw == "027") { *//kode admin* }else if (inputusn == user.username && inputpw == user.password) { *//kode pendaftar* }else {  percobaan--;  cout << "Login gagal! Sisa percobaan: " << percobaan << endl;  if (percobaan == 0) {  cout << "Anda telah gagal login 3 kali. Program dihentikan.\n";  exit(0);  }  } |

**D. Menu Admin**

Fitur ini digunakan untuk menampilkan menu admin, yang nantinya admin dapat memilih 1 dari 4 input tersedia.

**Source Code :**

cout << "\n|| ===== MENU ADMIN ===== ||\n";  
cout << "1. Tambah Workshop\n2. Lihat Workshop Tersedia\n3. Update Workshop\n4. Hapus Workshop\n6. Lihat Judul Terpanjang\n7. Logout\nMasukkan Pilihan: ";

cin >> pilihan;  
cin.ignore();  
*cout* *<<* "";

**E. Menambah Workshop (Create)**

Fitur admin untuk menambahkan workshop, admin meginput nama workshop, pemateri, tanggal, dan link grup WhatsApp.

**Source Code :**

**F. Menampilkan Workshop (Read)**

Fitur admin untuk menampilkan semua workshop yang tersedia ( sudah ditambahkan ) akan nampak nama workshop - pemateri - tanggal - dan link grup WhatsApp.

**Source Code :**

|  |
| --- |
| }else if (pilihan == 2){  if (dataprogram.totalWorkshops == 0) {  cout << "Belum ada Workshop tersedia" << endl;  } else {  cout << "Workshop yang tersedia : \n";  for (int i = 0; i < dataprogram.totalWorkshops; i++) {  cout << i + 1 << ". "  << dataprogram.workshop[i].judulworkshop << " - "  << dataprogram.workshop[i].pemateri << " - "  << dataprogram.workshop[i].tanggal << " - "  << dataprogram.workshop[i].linkgrup  << "\n";  }  } |

**G. Merubah Workshop (Update)**

Fitur ini digunakan admin untuk mengedit workshop yang sudah ada, admin dapat mengedit nama, pemateri, tanggal, serta grup WhatsApp

**Source Code :**

|  |
| --- |
| **}else if (pilihan == 3){  if (dataprogram.totalWorkshops == 0) {  cout << "Belum ada workshop untuk diubah." << endl;  } else {  for (int i = 0; i < dataprogram.totalWorkshops; i++) {  cout << i + 1 << ". "  << dataprogram.workshop[i].judulworkshop << " - "  << dataprogram.workshop[i].pemateri << " - "  << dataprogram.workshop[i].tanggal << " - "  << dataprogram.workshop[i].linkgrup  << "\n";  }  int pilih;  cout << "Masukkan nomor workshop yang akan diubah: ";  cin >> pilih;  if (pilih > 0 && pilih <= dataprogram.totalWorkshops) {  cin.ignore();  cout << "Masukkan nama Workshop baru: ";  getline(cin, dataprogram.workshop[pilih - 1].judulworkshop);  cout << "Masukkan nama pemateri baru: ";  getline(cin, dataprogram.workshop[pilih - 1].pemateri);  cout << "Masukkan tanggal baru: ";  getline(cin, dataprogram.workshop[pilih - 1].tanggal);  cout<< "Masukkan link grup baru: ";  getline(cin,dataprogram.workshop[pilih-1].linkgrup);  cout << "Workshop berhasil diubah." << endl;  } else {  cout << "Nomor workshop tidak valid." << endl;  }  }** |

**H. Menghapus Workshop (Delete)**

Fitur Admin yang digunakan untuk menghapus workshop yang sudah ada.

**Source Code :**

|  |
| --- |
| }else if (pilihan == 4){  if (dataprogram.totalWorkshops == 0) {  cout << "Belum ada workshop untuk dihapus." << endl;  } else {  for (int i = 0; i < dataprogram.totalWorkshops; i++) {  cout << i + 1 << ". "  << dataprogram.workshop[i].judulworkshop << " - "  << dataprogram.workshop[i].pemateri << " - "  << dataprogram.workshop[i].tanggal << " - "  << dataprogram.workshop[i].linkgrup  << "\n";  }   int input ;  cout << "Masukkan nomor workshop yang akan dihapus: ";  cin >> input;  if (input >= 1 && input <= dataprogram.totalWorkshops) {  for (int i = input - 1; i < dataprogram.totalWorkshops - 1; i++) {  dataprogram.workshop[i].judulworkshop = dataprogram.workshop[i + 1].judulworkshop;  dataprogram.workshop[i].pemateri = dataprogram.workshop[i + 1].pemateri;  dataprogram.workshop[i].tanggal = dataprogram.workshop[i + 1].tanggal;  dataprogram.workshop[i].linkgrup = dataprogram.workshop[i + 1].linkgrup;  }  dataprogram.totalWorkshops--;  cout << "Workshop berhasil dihapus." << endl;  } else {  cout << "Nomor workshop tidak valid." << endl;  }  }  } |

**I. Menu Pengguna**

Fitur untuk menampilkan menu untuk login sebagai pendaftar apabila sudah register, terdapat pilihan untuk melihat workshop, dan logout.

**Source Code :**

|  |
| --- |
| cout << "\n|| ===== MENU PENDAFTAR ===== ||\n";  do {  cout << "\n|| ===== WorkshopsToDo ===== ||\n";  cout << "1. Lihat Workshop Tersedia\n2. Logout\nPilih: ";  cin >> pilihan;  cin.ignore(); |

**J. Mendapat Link Grup**

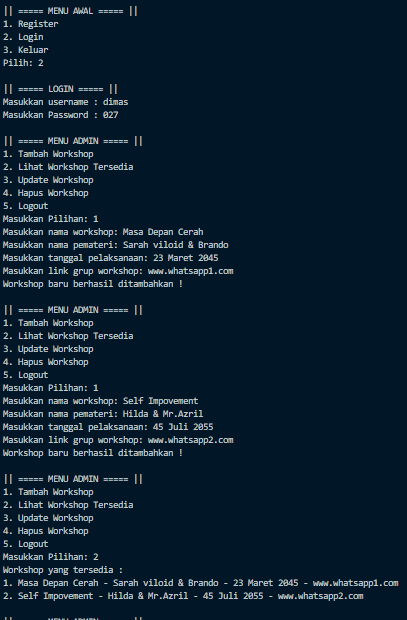
Fitur ini digunakan ketika pendaftar memilih salah satu workshop untuk bergabung.

**Source Code :**

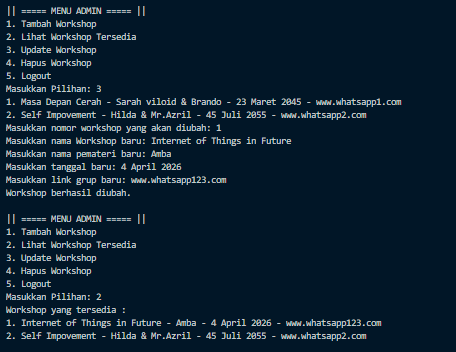
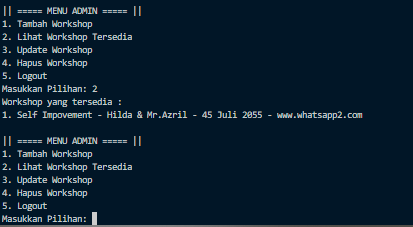
|  |
| --- |
| **int pilihanWorkshop;  cout << "Pilih nomor workshop untuk melihat link grup: ";  cin >> pilihanWorkshop;  cin.ignore();  if (pilihanWorkshop > 0 && pilihanWorkshop <= dataprogram.totalWorkshops) {  cout << "Link grup untuk " << dataprogram.workshop[pilihanWorkshop - 1].judulworkshop  << " adalah: " << dataprogram.workshop[pilihanWorkshop - 1].linkgrup << "\n";  cout<<"Silahkan join grup untuk mendapat info seputar Workshop\n";  } else {  cout << "Nomor workshop tidak valid." << endl;  }** |

# 

# 4. Uji Coba dan Hasil Output

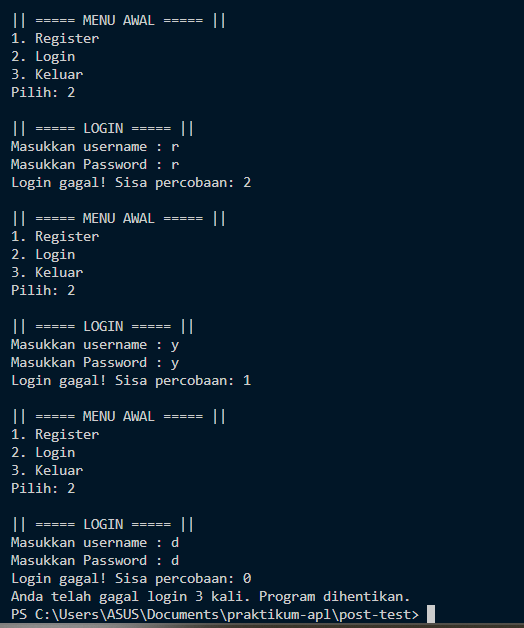
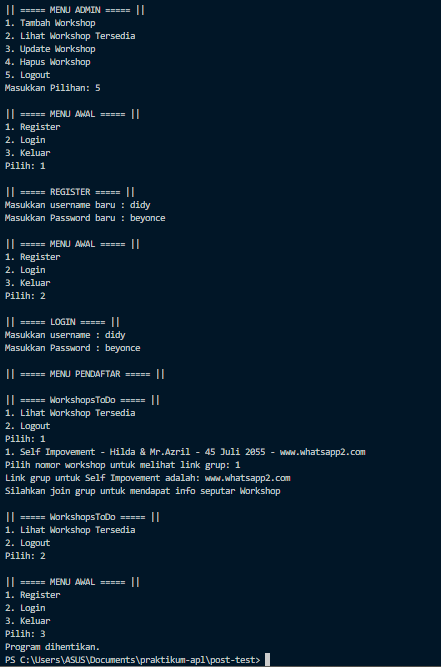
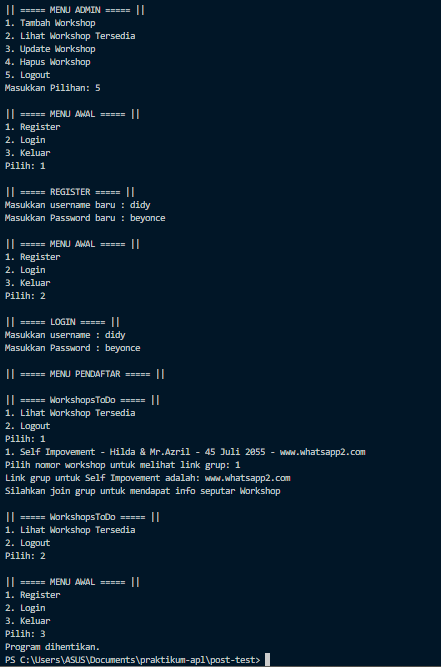
**

Gambar 4.1 Create dan Read pada Menu Admin

**

Gambar 4.3 Delete Workshop Menu Admin

Gambar 4.2 Update Workshop Menu Admin

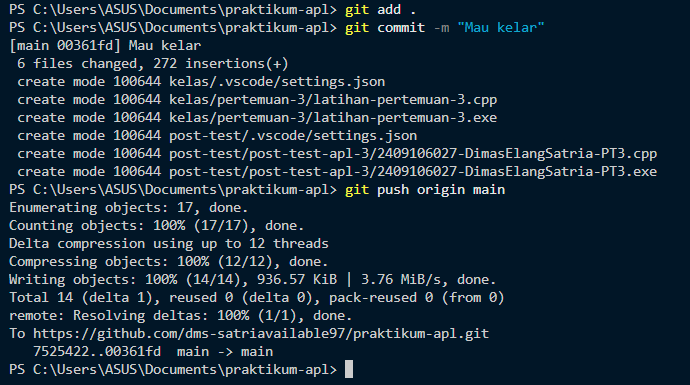
**

Gambar 4.5 Join Workshop Menu Pendaftar Hingga Keluar Program

Gambar 4.4 Logout Admin dan Login Pendaftar

Gambar 4.6 Login Gagal

# 5. Langkah-Langkah Git pada VSCode

Karena sebelumnya kita telah menginialisasi github dengan repository lokal, maka kita hanya perlu melakukan 3 langkah git yaitu git add ., git commit, dan git push.

Gambar 5.1 GitHub

1. Lakukan git add untuk menambahkan file apa saja yang ingin kita commit. Ketik perintah git add . (titik) pada terminal, untuk menambahkan semua file.
2. Lakukan git commit untuk membuat checkpoint. Ketik perintah git commit -m “Pesan commit” pada terminal.
3. Lakukan git push untuk meng-upload semua hal yang ada pada repository lokal ke Github. Ketik perintah git push -u origin main pada terminal.

Maka secara otomatis semua perubahan, entah menambahkan, mengedit, atau menghapus akan langsung dikenali di git, sehingga semua file sudah masuk di github.