

**LAPORAN PRAKTIKUM**  
**POSTTEST 6**  
**ALGORITMA PEMROGRAMAN LANJUT**

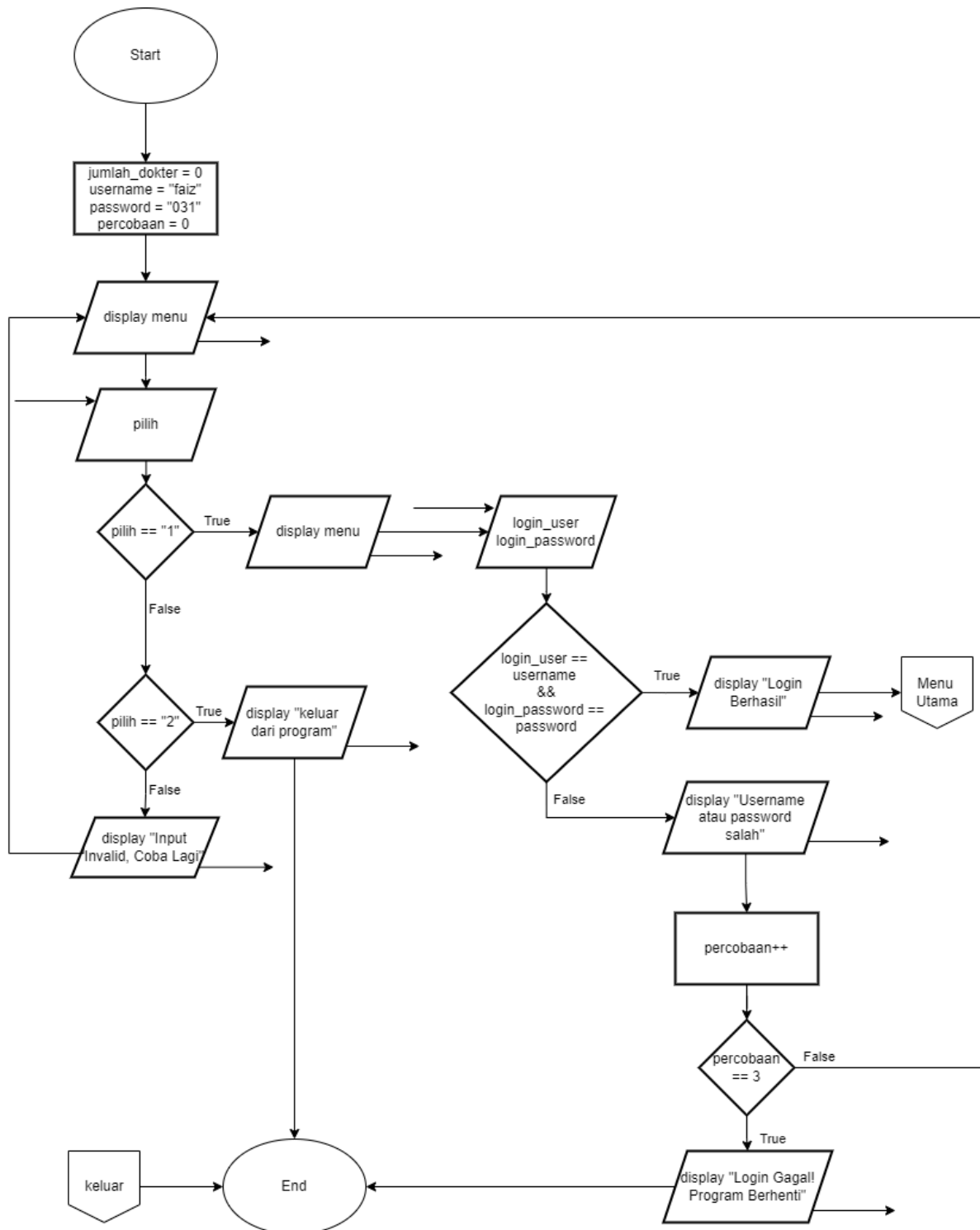


**Disusun oleh:**  
**Muhammad Faiz Lazuardi (2409106031)**  
**Kelas (A2 '24)**

**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**  
**2025**

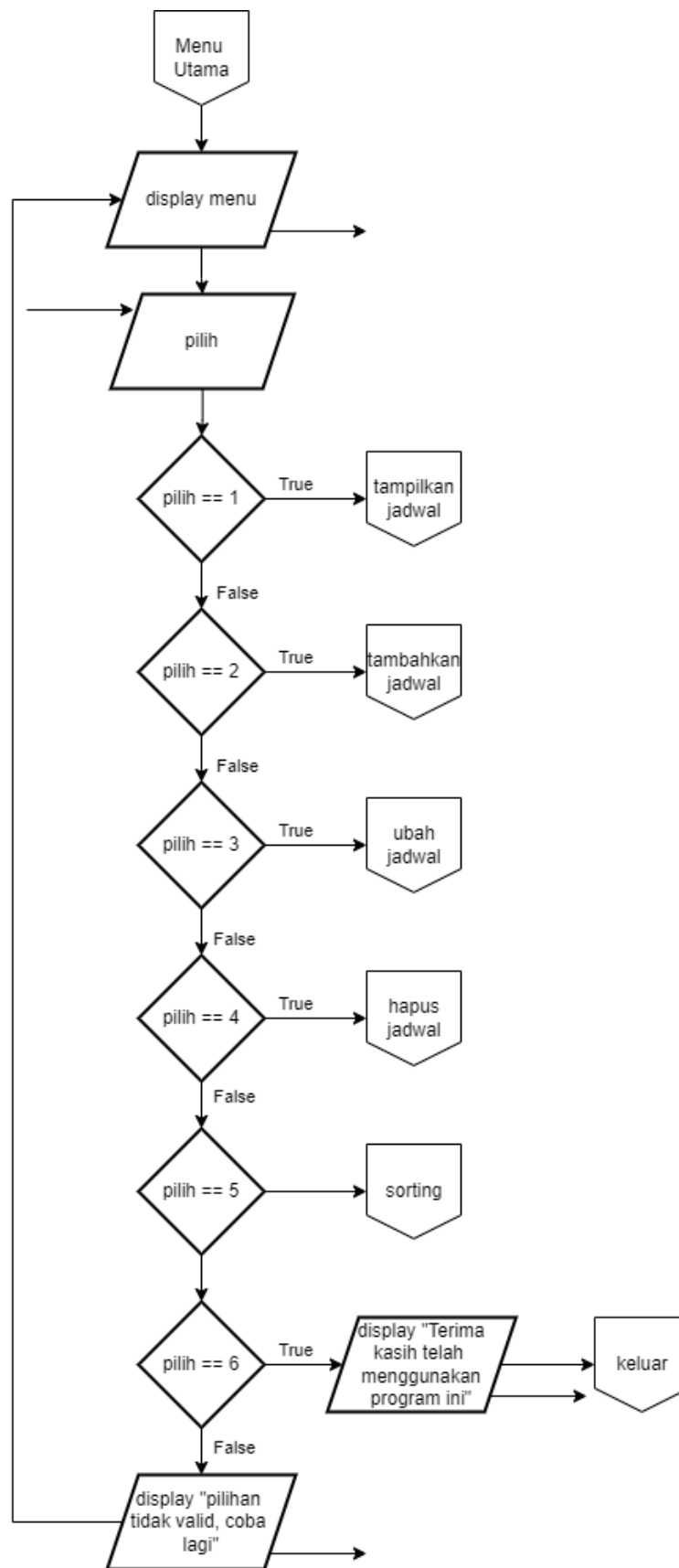
## 1. Flowchart

### 1.1 Menu Login



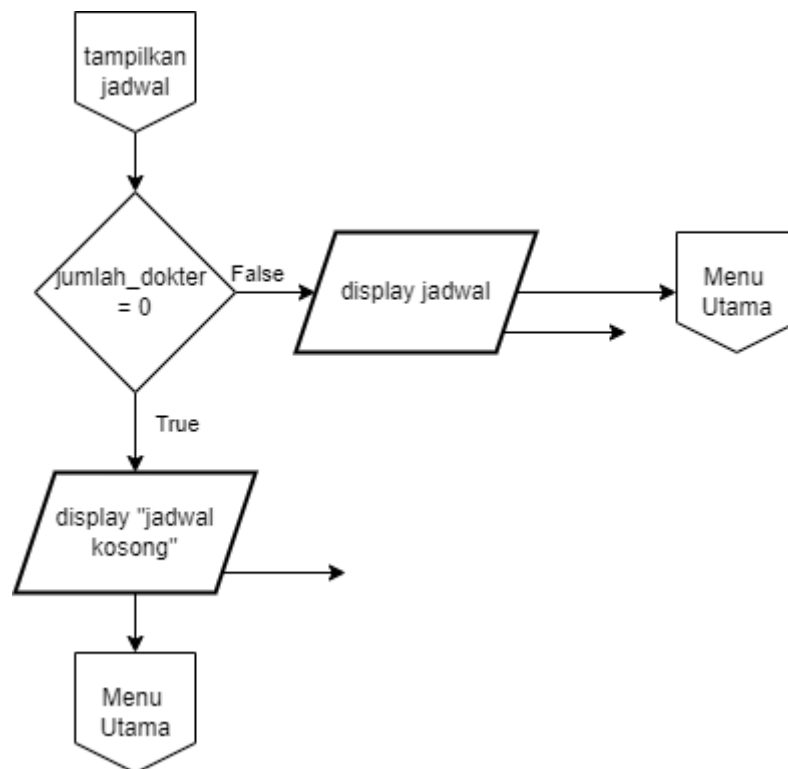
Gambar 1.1 Menu Login

## 1.2 Menu Utama



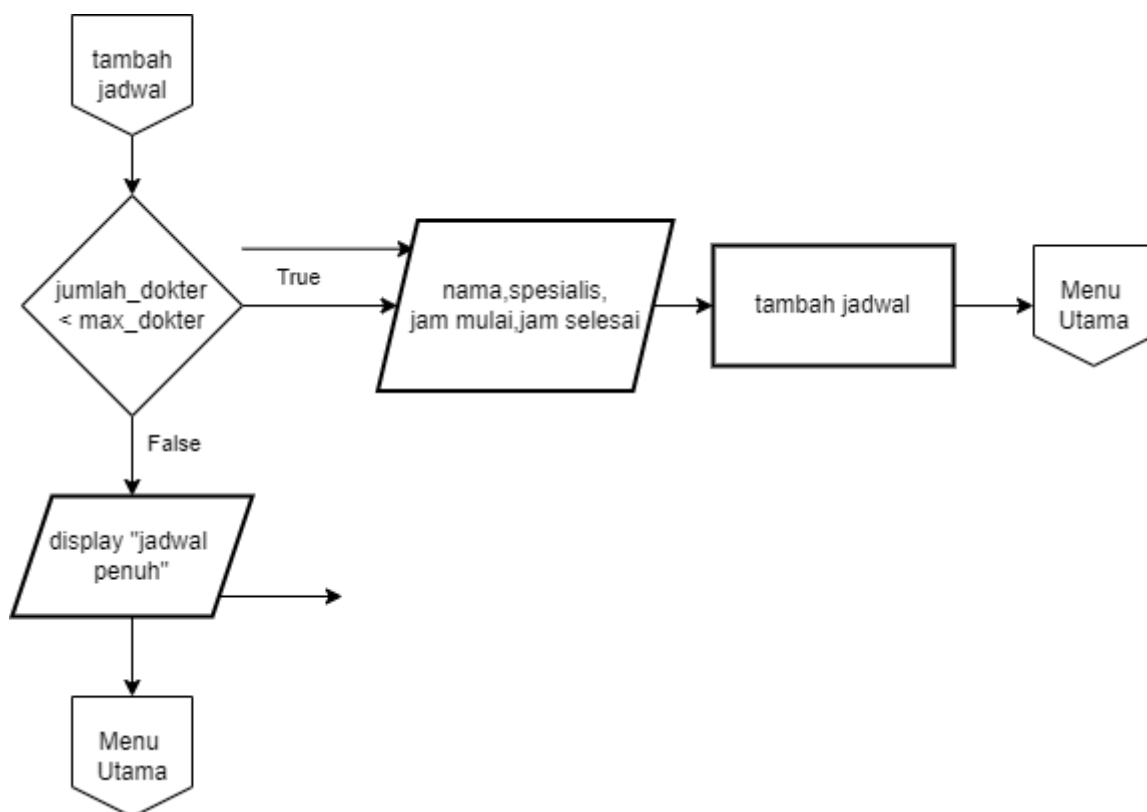
Gambar 1.2 Menu Utama

### 1.3 Menu Read



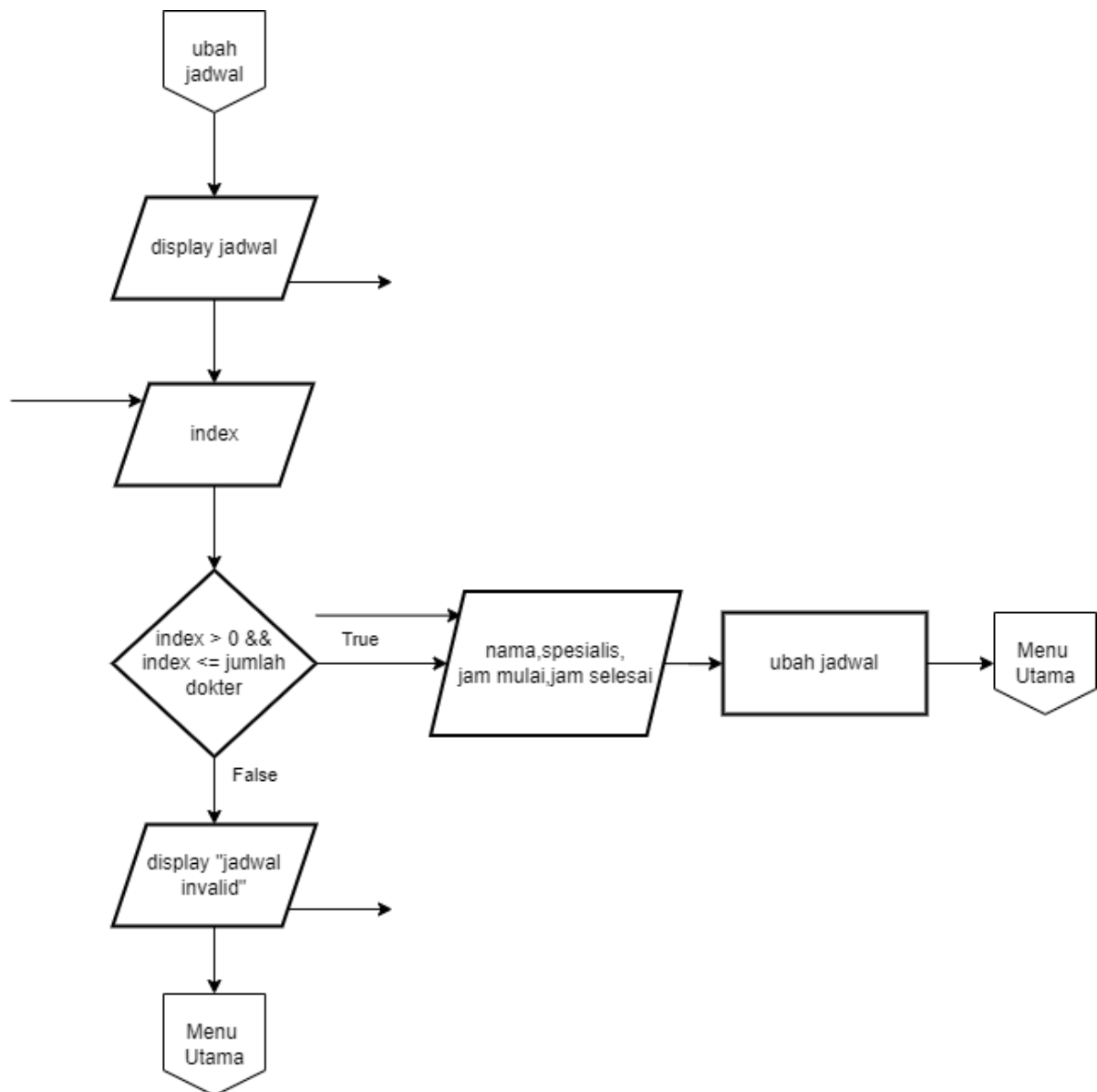
Gambar 1.3 Menu Read

### 1.4 Menu Create



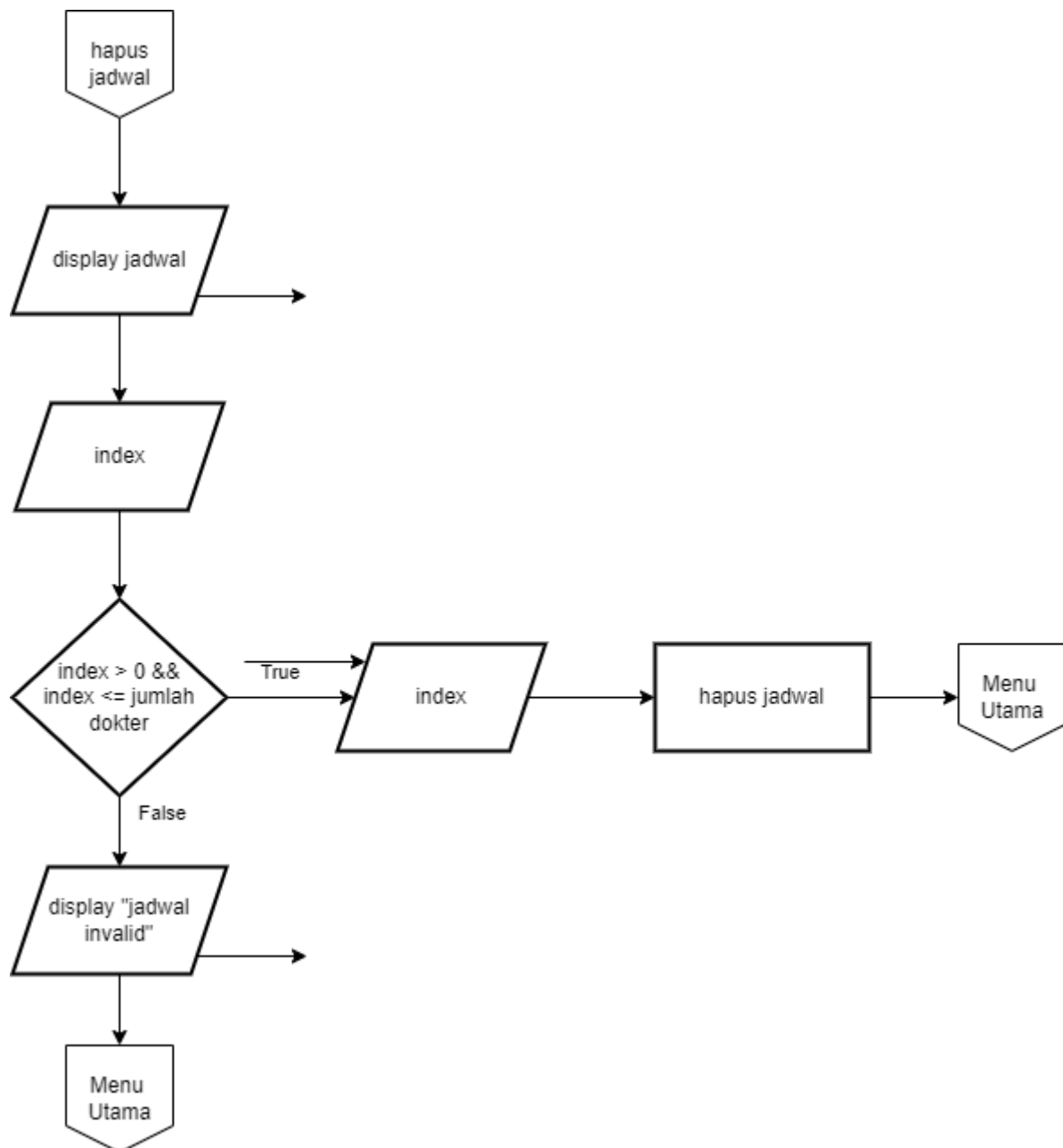
Gambar 1.4 Menu Create

## 1.5 Menu Update



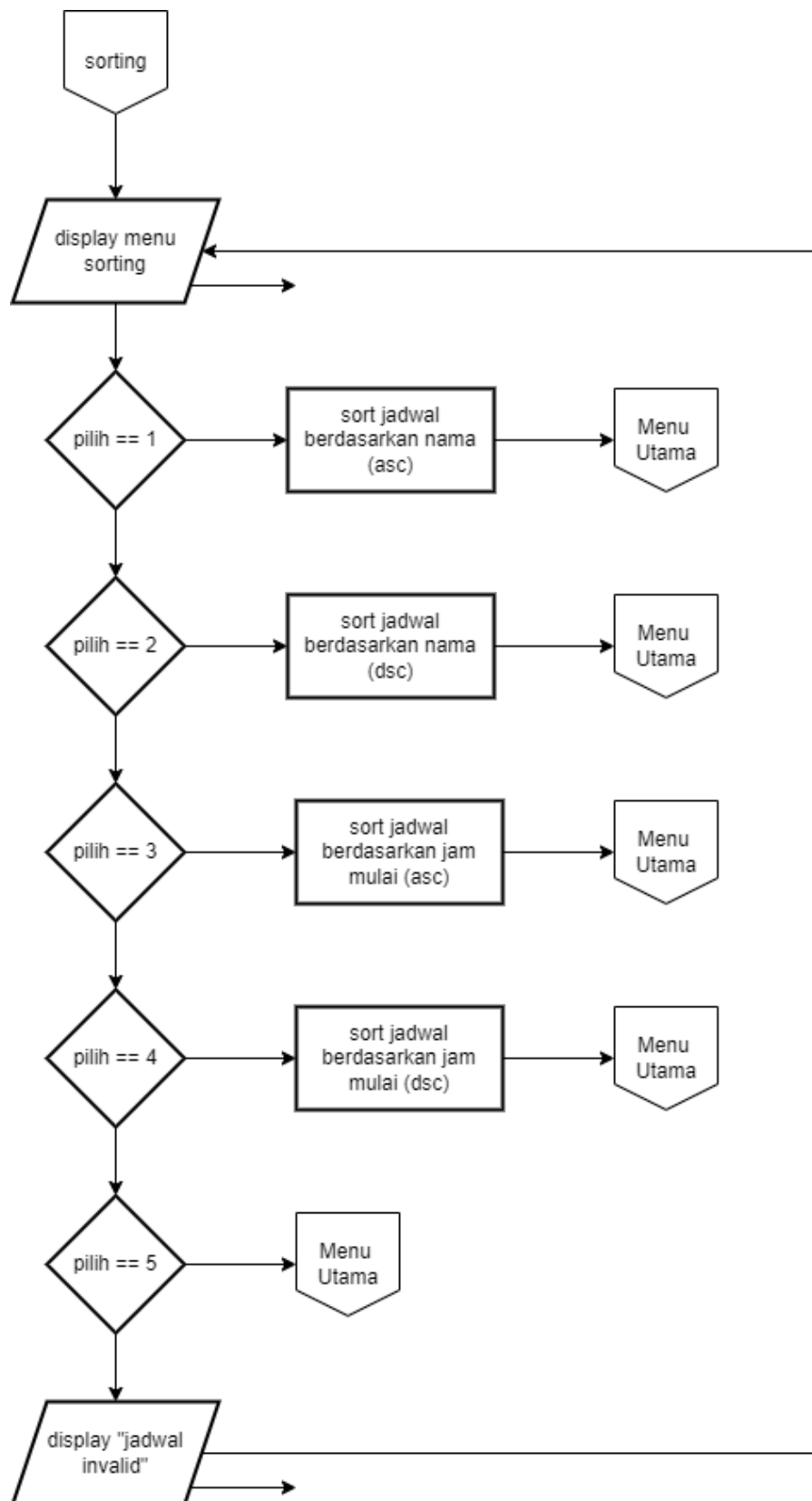
Gambar 1.5 Menu Update

## 1.6 Menu Delete



Gambar 1.6 Menu Delete

## 1.7 Sorting



Gambar 1.7 Menu Sorting

## 2. Analisis Program

Sebuah program yang berfungsi untuk manage jadwal praktek dokter spesialis.

## 3. Source Code

### A. Login

Fitur ini digunakan sebagai syarat untuk menjalankan program manajemen jadwal praktek dokter spesialis. Data yang diminta berupa username dan password.

Source Code:

```
void login() {
    int percobaan = 0;
    string username = "faiz", password = "031", login_user, login_password,
    pilih;
    bool login_berhasil = false;

    while (percobaan < 3) {
        cout << "\n<==== Login Page >====>\n";
        cout << "Username: "; cin >> login_user;
        cout << "Password: "; cin >> login_password;

        if (login_user == username and login_password == password) {
            cout << "Login berhasil!\n";
            login_berhasil = true;
            break;
        } else {
            cout << "Username atau password salah!\n";
            percobaan++;
        }
    }

    if (!login_berhasil){
        cout << "Login gagal! Program Berhenti.\n";
        login_gagal = true;
        return;
    }

    while (login_berhasil){
        menu_utama();
        cin >> pilih;
        cin.ignore();
        clear();
    }
}
```



```

        if (pilih == "1"){
            int index = 0;
            lihat_jadwal(index);
        } else if (pilih == "2"){
            tambah_jadwal();
        } else if (pilih == "3"){
            ubah_jadwal();
        } else if (pilih == "4"){
            hapus_jadwal();
        } else if (pilih == "5"){
            sorting_jadwal();
        } else if (pilih == "6"){
            login_berhasil = false;
        } else {
            cout << "Pilihan tidak valid! Coba lagi.\n";
        }
    }
}

```

## B. Menu Utama

Menu Utama berisi beberapa pilihan fitur. Pilihan 1 sampai 4 merupakan fitur CRUD dan pilihan 5 merupakan fitur untuk kembali ke halaman login.

### Source Code:

```

void menu_utama() {
    cout << "\n<===== Jadwal Praktek Dokter =====>\n";
    cout << "1. Tampilkan Jadwal\n";
    cout << "2. Tambah Jadwal\n";
    cout << "3. Ubah Jadwal\n";
    cout << "4. Hapus Jadwal\n";
    cout << "5. Logout\n";
    cout << "Pilih Fitur: ";
}

```

### C. Melihat Jadwal

Menampilkan jadwal.

#### Source Code:

```
bool lihat_jadwal(int index) {
    if (jumlah_dokter == 0) {
        cout << "Jadwal Kosong" << endl;
    }

    if (index == 0) {
        for(int i = 0; i < jumlah_dokter; ++i){
            cout << i + 1 << ". Nama: " << jadwal_praktek[i].nama_dokter <<
endl;
            cout << " " << "Spesialis: " << jadwal_praktek[i].spesialis <<
endl;
            cout << " " << "Jam Mulai: " << jadwal_praktek[i].jadwal.jam_mulai
<< endl;
            cout << " " << "Jam Selesai: " <<
jadwal_praktek[i].jadwal.jam_selesai << endl;
            cout << " " << endl;
            index++;
        }
    }

    if (index >= jumlah_dokter) {
        return true;
    }
    return lihat_jadwal(index + 1);
}
```

## D. Menambahkan Jadwal

Menambahkan jadwal.

### Source Code:

```
void tambah_jadwal() {
    if (jumlah_dokter >= MAX_DOKTER) {
        cout << "Jadwal Penuh.\n";
    }

    double jam_mulai, jam_selesai;
    string nama_dokter, spesialis;

    cout << "Masukkan Nama Dokter: ";
    cin >> nama_dokter;
    cout << "Masukkan Spesialis: ";
    cin >> spesialis;
    cin.ignore();
    cout << "Masukkan Jam Mulai: ";
    cin >> jam_mulai;
    cout << "Masukkan Jam Selesai: ";
    cin >> jam_selesai;

    if (tambah_jadwal(&nama_dokter, &spesialis) && tambah_jadwal(&jam_mulai,
&jam_selesai)) {
        cout << "Jadwal berhasil ditambahkan!\n";
    } else {
        cout << "Gagal menambahkan jadwal.\n";
    }
}

bool tambah_jadwal(string* nama_dokter, string* spesialis) {
    if (jumlah_dokter >= MAX_DOKTER) return false;
    jadwal_praktek[jumlah_dokter].nama_dokter = *nama_dokter;
    jadwal_praktek[jumlah_dokter].spesialis = *spesialis;
    return true;
}

bool tambah_jadwal(double* jam_mulai, double* jam_selesai) {
    if (jumlah_dokter >= MAX_DOKTER) return false;
    jadwal_praktek[jumlah_dokter].jadwal.jam_mulai = *jam_mulai;
    jadwal_praktek[jumlah_dokter].jadwal.jam_selesai = *jam_selesai;
    jumlah_dokter++;
    return true;
}
```

## E. Mengubah Jadwal

Mengubah jadwal.

### Source Code:

```
void ubah_jadwal() {
    if (jumlah_dokter == 0) {
        cout << "Jadwal Kosong" << endl;
    } else {
        int index = 0;
        lihat_jadwal(index);
        int pilih;
        cout << "Pilih Jadwal Yang Ingin Di Ubah: ";
        cin >> pilih;
        if (pilih > 0 && pilih <= jumlah_dokter) {
            cout << "Masukkan Nama Dokter baru: "; cin >> jadwal_praktek[pilih - 1].nama_dokter;
            cout << "Masukkan Spesialis baru: "; cin >> jadwal_praktek[pilih - 1].spesialis;
            cout << "Masukkan Jam Mulai baru: "; cin >> jadwal_praktek[pilih - 1].jadwal.jam_mulai;
            cout << "Masukkan Jam Selesai baru: "; cin >> jadwal_praktek[pilih - 1].jadwal.jam_selesai;
            cout << "Jadwal berhasil diperbarui!\n";

        } else {
            cout << "Jadwal Tidak Valid!\n";
        }
    }
}
```

## F. Menghapus Jadwal

Menghapus jadwal.

Source Code:

```
void hapus_jadwal() {
    if (jumlah_dokter == 0) {
        cout << "Jadwal Kosong" << endl;
    } else {
        int index = 0;
        lihat_jadwal(index);
        int pilih;
        cout << "Pilih Jadwal Yang Ingin Di Hapus: ";
        cin >> pilih;
        if (pilih > 0 && pilih <= jumlah_dokter) {
            for (int i = pilih - 1; i < jumlah_dokter - 1; i++) {
                jadwal_praktek[i] = jadwal_praktek[i + 1];
            }
            jumlah_dokter--;
            cout << "Jadwal berhasil dihapus!\n";
        } else {
            cout << "Jadwal Tidak Valid!\n";
        }
    }
}
```

## G. Sorting

Mengurutkan Jadwal.

Source Code:

```
void sorting_jadwal() {
    while (true) {
        if (jumlah_dokter == 0) {
            cout << "Jadwal Kosong" << endl;
            return;
        }

        cout << "<==== Menu Sorting =====>\n";
        cout << "1. Berdasarkan Nama ascending\n";
        cout << "2. Berdasarkan Nama Descending\n";
        cout << "3. Berdasarkan Jam Mulai ascending\n";
    }
}
```

```

    cout << "4. Berdasarkan Jam Mulai Descending\n";
    cout << "5. Kembali\n";
    cout << "Pilih Fitur: ";

    int pilih;
    cin >> pilih;
    cin.ignore();
    clear();

    switch (pilih) {
        case 1:
            sort_nama(jadwal_praktek, 0, jumlah_dokter - 1, true);
            cout << "Produk diurutkan berdasarkan nama (ascending).\n";
            break;
        case 2:
            sort_nama(jadwal_praktek, 0, jumlah_dokter - 1, false);
            cout << "Produk diurutkan berdasarkan nama (descending).\n";
            break;
        case 3:
            sort_jam(jadwal_praktek, 0, jumlah_dokter - 1, true);
            cout << "Produk diurutkan berdasarkan Jam Mulai (ascending).\n";
            break;
        case 4:
            sort_jam(jadwal_praktek, 0, jumlah_dokter - 1, false);
            cout << "Produk diurutkan berdasarkan Jam Mulai
(descending).\n";
            break;
        case 5:
            clear();
            return;
        default:
            cout << "Pilihan tidak valid!\n";
            continue;
    }
    return;
}

void tukar_jadwal(jadwal_dokter &a, jadwal_dokter&b) {
    jadwal_dokter temp = a;
    a = b;
    b = temp;
}

void sort_jam(jadwal_dokter arr[], int low, int high, bool asc) {
    if (low < high) {
        int pivot = low;
        int i = low;
        int j = high;
        while (i <= j) {
            if (asc) {

```

```

        while (arr[i].jadwal.jam_mulai <= arr[pivot].jadwal.jam_mulai &&
i <= high) i++;
        while (arr[j].jadwal.jam_mulai > arr[pivot].jadwal.jam_mulai &&
j >= low) j--;
    } else {
        while (arr[i].jadwal.jam_mulai >= arr[pivot].jadwal.jam_mulai &&
i <= high) i++;
        while (arr[j].jadwal.jam_mulai < arr[pivot].jadwal.jam_mulai &&
j >= low) j--;
    }
    if (i < j) tukar_jadwal(arr[i], arr[j]);
}
tukar_jadwal(arr[j], arr[pivot]);
sort_jam(arr, low, j - 1, asc);
sort_jam(arr, j + 1, high, asc);
}
}

void sort_nama(jadwal_dokter arr[], int low, int high, bool asc) {
    if (low < high) {
        int pivot = low;
        int i = low;
        int j = high;
        while (i <= j) {
            if (asc) {
                while (arr[i].nama_dokter <= arr[pivot].nama_dokter && i <=
high) i++;
                while (arr[j].nama_dokter > arr[pivot].nama_dokter && j >= low)
j--;
            } else {
                while (arr[i].nama_dokter >= arr[pivot].nama_dokter && i <=
high) i++;
                while (arr[j].nama_dokter < arr[pivot].nama_dokter && j >= low)
j--;
            }
            if (i < j) tukar_jadwal(arr[i], arr[j]);
        }
        tukar_jadwal(arr[j], arr[pivot]);
        sort_nama(arr, low, j - 1, asc);
        sort_nama(arr, j + 1, high, asc);
    }
}
}

```

## 4. Screenshot Program

```

1  ● ● ●
2  #include <fstream>
3  #include <string>
4  #include <iomanip>
5
6  using namespace std;
7
8  const int N = 1000000;
9  const int M = 1000000;
10 const int K = 1000000;
11
12 int main() {
13     fstream fin("input.txt");
14     fstream fout("output.txt");
15     if (!fin.is_open() || !fout.is_open()) {
16         return 1;
17     }
18     int n, m, k;
19     fin >> n >> m >> k;
20     int a[N], b[M], c[K];
21     for (int i = 0; i < n; i++) {
22         fin >> a[i];
23     }
24     for (int i = 0; i < m; i++) {
25         fin >> b[i];
26     }
27     for (int i = 0; i < k; i++) {
28         fin >> c[i];
29     }
30     int ans = 0;
31     for (int i = 0; i < n; i++) {
32         for (int j = 0; j < m; j++) {
33             for (int l = 0; l < k; l++) {
34                 if (a[i] + b[j] + c[l] == 0) {
35                     ans++;
36                 }
37             }
38         }
39     }
40     fout << ans;
41     return 0;
42 }

```

#### Gambar 4.1 Code Program



[illegible]

### Gambar 4.2 Code Program

```

101 while (true) {
102     if (true) {
103         // ...
104     }
105     // ...
106 }
107
108 while (true) {
109     if (true) {
110         // ...
111     }
112     // ...
113 }
114
115 while (true) {
116     if (true) {
117         // ...
118     }
119     // ...
120 }
121
122 while (true) {
123     if (true) {
124         // ...
125     }
126     // ...
127 }
128
129 while (true) {
130     if (true) {
131         // ...
132     }
133     // ...
134 }
135
136 while (true) {
137     if (true) {
138         // ...
139     }
140     // ...
141 }
142
143 while (true) {
144     if (true) {
145         // ...
146     }
147     // ...
148 }
149
150 while (true) {
151     if (true) {
152         // ...
153     }
154     // ...
155 }
156
157 while (true) {
158     if (true) {
159         // ...
160     }
161     // ...
162 }
163
164 while (true) {
165     if (true) {
166         // ...
167     }
168     // ...
169 }
170
171 while (true) {
172     if (true) {
173         // ...
174     }
175     // ...
176 }
177
178 while (true) {
179     if (true) {
180         // ...
181     }
182     // ...
183 }
184
185 while (true) {
186     if (true) {
187         // ...
188     }
189     // ...
190 }
191
192 while (true) {
193     if (true) {
194         // ...
195     }
196     // ...
197 }
198
199 while (true) {
200     if (true) {
201         // ...
202     }
203     // ...
204 }
205
206 while (true) {
207     if (true) {
208         // ...
209     }
210     // ...
211 }
212
213 while (true) {
214     if (true) {
215         // ...
216     }
217     // ...
218 }
219
220 while (true) {
221     if (true) {
222         // ...
223     }
224     // ...
225 }
226
227 while (true) {
228     if (true) {
229         // ...
230     }
231     // ...
232 }
233
234 while (true) {
235     if (true) {
236         // ...
237     }
238     // ...
239 }
240
241 while (true) {
242     if (true) {
243         // ...
244     }
245     // ...
246 }
247
248 while (true) {
249     if (true) {
250         // ...
251     }
252     // ...
253 }
254
255 while (true) {
256     if (true) {
257         // ...
258     }
259     // ...
260 }
261
262 while (true) {
263     if (true) {
264         // ...
265     }
266     // ...
267 }
268
269 while (true) {
270     if (true) {
271         // ...
272     }
273     // ...
274 }
275
276 while (true) {
277     if (true) {
278         // ...
279     }
280     // ...
281 }
282
283 while (true) {
284     if (true) {
285         // ...
286     }
287     // ...
288 }
289
290 while (true) {
291     if (true) {
292         // ...
293     }
294     // ...
295 }
296
297 while (true) {
298     if (true) {
299         // ...
300     }
301     // ...
302 }
303
304 while (true) {
305     if (true) {
306         // ...
307     }
308     // ...
309 }
310
311 while (true) {
312     if (true) {
313         // ...
314     }
315     // ...
316 }
317
318 while (true) {
319     if (true) {
320         // ...
321     }
322     // ...
323 }
324
325 while (true) {
326     if (true) {
327         // ...
328     }
329     // ...
330 }
331
332 while (true) {
333     if (true) {
334         // ...
335     }
336     // ...
337 }
338
339 while (true) {
340     if (true) {
341         // ...
342     }
343     // ...
344 }
345
346 while (true) {
347     if (true) {
348         // ...
349     }
350     // ...
351 }
352
353 while (true) {
354     if (true) {
355         // ...
356     }
357     // ...
358 }
359
360 while (true) {
361     if (true) {
362         // ...
363     }
364     // ...
365 }
366
367 while (true) {
368     if (true) {
369         // ...
370     }
371     // ...
372 }
373
374 while (true) {
375     if (true) {
376         // ...
377     }
378     // ...
379 }
380
381 while (true) {
382     if (true) {
383         // ...
384     }
385     // ...
386 }
387
388 while (true) {
389     if (true) {
390         // ...
391     }
392     // ...
393 }
394
395 while (true) {
396     if (true) {
397         // ...
398     }
399     // ...
400 }
401
402 while (true) {
403     if (true) {
404         // ...
405     }
406     // ...
407 }
408
409 while (true) {
410     if (true) {
411         // ...
412     }
413     // ...
414 }
415
416 while (true) {
417     if (true) {
418         // ...
419     }
420     // ...
421 }
422
423 while (true) {
424     if (true) {
425         // ...
426     }
427     // ...
428 }
429
430 while (true) {
431     if (true) {
432         // ...
433     }
434     // ...
435 }
436
437 while (true) {
438     if (true) {
439         // ...
440     }
441     // ...
442 }
443
444 while (true) {
445     if (true) {
446         // ...
447     }
448     // ...
449 }
450
451 while (true) {
452     if (true) {
453         // ...
454     }
455     // ...
456 }
457
458 while (true) {
459     if (true) {
460         // ...
461     }
462     // ...
463 }
464
465 while (true) {
466     if (true) {
467         // ...
468     }
469     // ...
470 }
471
472 while (true) {
473     if (true) {
474         // ...
475     }
476     // ...
477 }
478
479 while (true) {
480     if (true) {
481         // ...
482     }
483     // ...
484 }
485
486 while (true) {
487     if (true) {
488         // ...
489     }
490     // ...
491 }
492
493 while (true) {
494     if (true) {
495         // ...
496     }
497     // ...
498 }
499
500 while (true) {
501     if (true) {
502         // ...
503     }
504     // ...
505 }
506
507 while (true) {
508     if (true) {
509         // ...
510     }
511     // ...
512 }
513
514 while (true) {
515     if (true) {
516         // ...
517     }
518     // ...
519 }
520
521 while (true) {
522     if (true) {
523         // ...
524     }
525     // ...
526 }
527
528 while (true) {
529     if (true) {
530         // ...
531     }
532     // ...
533 }
534
535 while (true) {
536     if (true) {
537         // ...
538     }
539     // ...
540 }
541
542 while (true) {
543     if (true) {
544         // ...
545     }
546     // ...
547 }
548
549 while (true) {
550     if (true) {
551         // ...
552     }
553     // ...
554 }
555
556 while (true) {
557     if (true) {
558         // ...
559     }
560     // ...
561 }
562
563 while (true) {
564     if (true) {
565         // ...
566     }
567     // ...
568 }
569
570 while (true) {
571     if (true) {
572         // ...
573     }
574     // ...
575 }
576
577 while (true) {
578     if (true) {
579         // ...
580     }
581     // ...
582 }
583
584 while (true) {
585     if (true) {
586         // ...
587     }
588     // ...
589 }
590
591 while (true) {
592     if (true) {
593         // ...
594     }
595     // ...
596 }
597
598 while (true) {
599     if (true) {
600         // ...
601     }
602     // ...
603 }
604
605 while (true) {
606     if (true) {
607         // ...
608     }
609     // ...
610 }
611
612 while (true) {
613     if (true) {
614         // ...
615     }
616     // ...
617 }
618
619 while (true) {
620     if (true) {
621         // ...
622     }
623     // ...
624 }
625
626 while (true) {
627     if (true) {
628         // ...
629     }
630     // ...
631 }
632
633 while (true) {
634     if (true) {
635         // ...
636     }
637     // ...
638 }
639
640 while (true) {
641     if (true) {
642         // ...
643     }
644     // ...
645 }
646
647 while (true) {
648     if (true) {
649         // ...
650     }
651     // ...
652 }
653
654 while (true) {
655     if (true) {
656         // ...
657     }
658     // ...
659 }
660
661 while (true) {
662     if (true) {
663         // ...
664     }
665     // ...
666 }
667
668 while (true) {
669     if (true) {
670         // ...
671     }
672     // ...
673 }
674
675 while (true) {
676     if (true) {
677         // ...
678     }
679     // ...
680 }
681
682 while (true) {
683     if (true) {
684         // ...
685     }
686     // ...
687 }
688
689 while (true) {
690     if (true) {
691         // ...
692     }
693     // ...
694 }
695
696 while (true) {
697    
```

### Gambar 4.3 Code Program

## 5. Git

### 5.1 Git Init

Git init merupakan command yang berfungsi untuk menginisiasi repository yang ada pada file lokal dan berlokasi di folder .git. git branch -m master main merupakan command yang berfungsi untuk merubah branch repository dari master ke main.

```
ASUS@DESKTOP-14RFNS8 MINGW64 ~/Desktop
$ mkdir praktikum-apl

ASUS@DESKTOP-14RFNS8 MINGW64 ~/Desktop
$ cd praktikum-apl/

ASUS@DESKTOP-14RFNS8 MINGW64 ~/Desktop/praktikum-apl
$ pwd
/c/Users/ASUS/Desktop/praktikum-apl

ASUS@DESKTOP-14RFNS8 MINGW64 ~/Desktop/praktikum-apl
$ git init
Initialized empty Git repository in C:/Users/ASUS/Desktop/praktikum-apl/.git/

ASUS@DESKTOP-14RFNS8 MINGW64 ~/Desktop/praktikum-apl (master)
$ git branch -m master main

ASUS@DESKTOP-14RFNS8 MINGW64 ~/Desktop/praktikum-apl (main)
```

Gambar 5.1 Git init dan Change branch

### 5.2 Git Add

Menambahkan file yang ingin di commit.

```
ASUS@DESKTOP-14RFNS8 MINGW64 ~/Desktop/praktikum-apl/post-test (main)
$ git add .
```

Gambar 5.2 Membuat File dan Git add

### 5.3 Git Commit

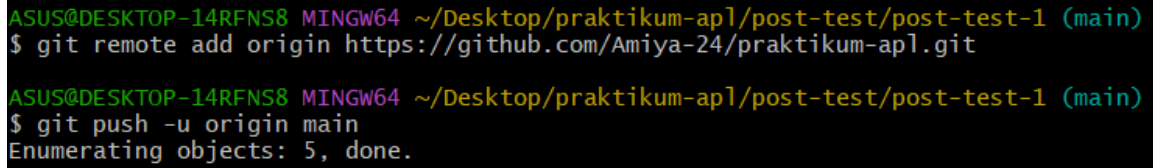
Menyimpan perubahan yang disimpan pada folder lokal dengan sebuah pesan.

```
ASUS@DESKTOP-14RFNS8 MINGW64 ~/Desktop/praktikum-apl/post-test/post-test-6 (main)
$ git commit -m "Finish Post Test 6"
[main 8f766b8] Finish Post Test 6
2 files changed, 6 insertions(+), 6 deletions(-)
```

Gambar 5.3 Git Commit

## 5.4 Git Remote dan Git Push

Git remote berfungsi untuk menghubungkan repository yang ada di lokal dengan repository cloud pada github. Git push berfungsi untuk mengupload semua yang ada di lokal ke github.

A screenshot of a Windows terminal window with a black background and green text. The prompt is 'ASUS@DESKTOP-14RFNS8 MINGW64 ~'. The first command is 'git remote add origin https://github.com/Amiya-24/praktikum-apl.git'. The second command is 'git push -u origin main'. The output of the second command is 'Enumerating objects: 5, done.'.

```
ASUS@DESKTOP-14RFNS8 MINGW64 ~/Desktop/praktikum-apl/post-test/post-test-1 (main)
$ git remote add origin https://github.com/Amiya-24/praktikum-apl.git

ASUS@DESKTOP-14RFNS8 MINGW64 ~/Desktop/praktikum-apl/post-test/post-test-1 (main)
$ git push -u origin main
Enumerating objects: 5, done.
```

Gambar 5.4 Git Remote dan Git Push