

LAPORAN PRAKTIKUM
POSTTEST 7
ALGORITMA PEMROGRAMAN DASAR



Disusun oleh:

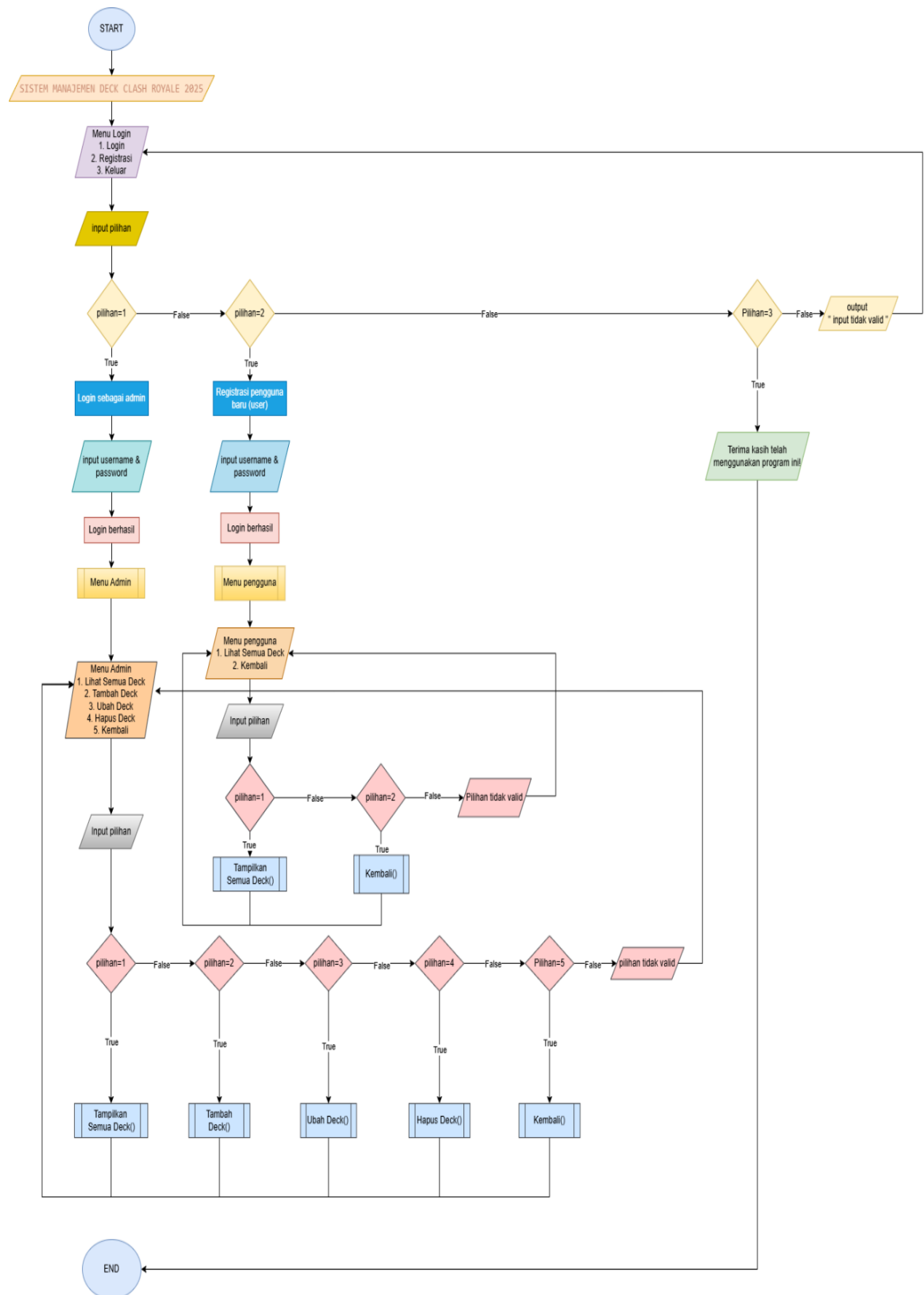
Navtaly Juman (2509106109)

Kelas (C1 '25)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA

2025

1.Flowchart



Gambar 1.1 Flowchart

2.Deskripsi Singkat Program

Program Sistem Manajemen Deck Clash Royale 2025 ini dibuat untuk mengelola berbagai deck pemain profesional Clash Royale. Program memiliki dua jenis pengguna, yaitu Admin dan User. Admin dapat melihat, menambah, mengubah, dan menghapus deck, sedangkan User hanya dapat melihat daftar deck. Program ini menggunakan fungsi dengan dan tanpa parameter, prosedur, serta menerapkan rekursi untuk navigasi menu dan error handling agar input pengguna tetap aman. Program ini juga menggunakan variabel global untuk data pengguna dan deck, serta variabel lokal dalam fungsi pengelolaan deck.

3. Source Code

A. Variabel Global

```
1  users = {
2    "Navtaly Juman": {"password": "2509106109", "role": "Admin"}
3  }
4
5  decks = {
6    1: {
7      "nama": "Miner Control",
8      "pemain": "Morten",
9      "strategi": "Control",
10     "kartu": ["Miner", "Poison", "Wall Breaker", "Skeletons", "Bats", "Valkyrie", "Bomb Tower", "Log"]
11   },
12   2: {
13     "nama": "X-Bow 2.9 Cycle",
14     "pemain": "Aragon",
15     "strategi": "Siege",
16     "kartu": ["X-Bow", "Tesla", "Ice Spirit", "Skeletons", "Ice Golem", "Fireball", "Log", "Archers"]
17   },
18   3: {
19     "nama": "LavaLoon",
20     "pemain": "Surgical Goblin",
21     "strategi": "Air Beatdown",
22     "kartu": ["Lava Hound", "Balloon", "Mega Minion", "Tombstone", "Minions", "Fireball", "Zap", "Miner"]
23   },
24   4: {
25     "nama": "Royal Giant Lightning",
26     "pemain": "Mohamed Light",
27     "strategi": "Beatdown",
28     "kartu": ["Royal Giant", "Fisherman", "Lightning", "Mother Witch", "Electro Spirit", "Skeletons", "Hunter", "Log"]
29   },
30   5: {
31     "nama": "Bridge Spam Pekka",
32     "pemain": "Ruben",
33     "strategi": "Control",
34     "kartu": ["Bandit", "Battle Ram", "Royal Ghost", "Dark Prince", "Magic Archer", "Pekka", "Poison", "Zap"]
35   }
36 }
```

Gambar 3.1 Variabel Global

B. Fungsi Tanpa Parameter

```

1
2 def tampilkan_deck():
3     if len(decks) == 0:
4         print("Belum ada deck yang ditambahkan.")
5     else:
6         for key, deck in decks.items():
7             print(f"\nID: {key}")
8             print(f>Nama Deck: {deck['nama']}")
9             print(f>Pemain: {deck['pemain']}")
10            print(f>Strategi: {deck['strategi']}")
11            print(f>Kartu: {deck['kartu']}")
12

```

Gambar 3.2 Fungsi Tanpa Parameter

C. Fungsi Dengan Parameter

```

1 def tambah_deck(jumlah_kartu):
2     nama = input("Nama Deck: ")
3     pemain = input("Nama Pemain: ")
4     strategi = input("Jenis Strategi: ")

```

Gambar 3.3 Fungsi Dengan Parameter

D. Variabel Lokal

```

1 kartu = []
2 print(f>Masukkan {jumlah_kartu} kartu:")
3 for i in range(jumlah_kartu):
4     kartu_baru = input(f>Kartu ke-{i+1}: ")
5     kartu += [kartu_baru]
6 new_id = len(decks) + 1
7 decks[new_id] = {"nama": nama, "pemain": pemain, "strategi": strategi, "kartu": kartu}
8 print("Deck berhasil ditambahkan!")

```

Gambar 3.4 Variabel Lokal

E. Fungsi Login & Register (Multiuser)

```
1 def login():
2     username = input("Username: ")
3     password = input("Password: ")
4     if username in users and users[username]["password"] == password:
5         print(f"\nSelamat datang {username}!\n")
6         return users[username]["role"], username
7     else:
8         print("Login gagal! Username atau password salah.")
9         return None, None
10 def register():
11     username = input("Masukkan Username: ")
12     password = input("Masukkan Password: ")
13     if username in users:
14         print("Username sudah digunakan!")
15     else:
16         users[username] = {"password": password, "role": "User"}
17         print("Akun berhasil dibuat!")
18
```

Gambar 3.5 Fungsi Login & Register (Multiuser)

F. Fungsi CRUD Deck

```
1 def hapus_deck(id_hapus):
2     if id_hapus in decks:
3         del decks[id_hapus]
4         print("Deck berhasil dihapus!")
5     else:
6         print("ID deck tidak ditemukan.")
7
8 def ubah_deck(id_ubah):
9     if id_ubah not in decks:
10        print("ID deck tidak ditemukan.")
11        return
12    deck = decks[id_ubah]
13    print(f"\nMengubah deck: {deck['nama']}")
14
15    deck["nama"] = input(f>Nama baru ({deck['nama']}): ") or deck["nama"]
16    deck["pemain"] = input(f>Pemain baru ({deck['pemain']}): ") or deck["pemain"]
17    deck["strategi"] = input(f>Strategi baru ({deck['strategi']}): ") or deck["strategi"]
18
19    print("Masukkan kartu baru (kosongkan jika tidak ingin mengubah):")
20    kartu_baru = []
21    for i in range(8):
22        k = input(f>Kartu ke-{i+1} ({deck['kartu'][i]}): ")
23        if k:
24            kartu_baru += [k]
25        else:
26            kartu_baru += [deck["kartu"][i]]
27    deck["kartu"] = kartu_baru
28    print("Deck berhasil diubah!")
```

Gambar 3.6 Fungsi CRUD Deck

G. Prosedur: Menu Admin, Error Handling (try-except), Rekursif

```
1  def menu_admin():
2      print("\n=== MENU ADMIN ===")
3      print("1. Lihat Semua Deck")
4      print("2. Tambah Deck")
5      print("3. Ubah Deck")
6      print("4. Hapus Deck")
7      print("5. Logout")
8
9      try:
10         pilihan = int(input("Pilih menu: "))
11     except ValueError:
12         print("Input harus berupa angka!")
13         return menu_admin()
14
15     if pilihan == 1:
16         tampilkan_deck()
17     elif pilihan == 2:
18         tambah_deck(8)
19     elif pilihan == 3:
20         try:
21             id_ubah = int(input("Masukkan ID deck yang ingin diubah: "))
22             ubah_deck(id_ubah)
23         except ValueError:
24             print("ID harus angka!")
25     elif pilihan == 4:
26         try:
27             id_hapus = int(input("Masukkan ID deck yang ingin dihapus: "))
28             hapus_deck(id_hapus)
29         except ValueError:
30             print("Input ID harus angka!")
31     elif pilihan == 5:
32         return
33     else:
34         print("Pilihan tidak valid!")
35
36     menu_admin()
```

Gambar 3.7 Prosedur: Menu Admin, Error Handling (try-except), Rekursif

H. Prosedur: Menu User, Fungsi Utama

```

1  def menu_user():
2      print("\n=== MENU USER ===")
3      print("1. Lihat Semua Deck")
4      print("2. Logout")
5      pilihan = input("Pilih menu: ")
6      if pilihan == "1":
7          tampilkan_deck()
8          menu_user()
9      elif pilihan == "2":
10         print("Logout berhasil.")
11     else:
12         print("Pilihan tidak valid!")
13         menu_user()
14
15 def main():
16     print("\n=== SISTEM MANAJEMEN DECK CLASH ROYALE 2025 ===")
17     print("=====")
18
19     while True:
20         print("1. Login")
21         print("2. Register")
22         print("3. Keluar")
23         pilihan = input("Pilih 1, 2, atau 3: ")
24
25         if pilihan == "1":
26             role, nama_user = login()
27             if role == "Admin":
28                 menu_admin()
29             elif role == "User":
30                 menu_user()
31         elif pilihan == "2":
32             register()
33         elif pilihan == "3":
34             print("Terima kasih telah menggunakan program ini!")
35             break
36         else:
37             print("Pilihan tidak valid!")
38     main()

```

Gambar 3.8 . Prosedur: Menu User, Fungsi Utama

4. Hasil Output

```

=== SISTEM MANAJEMEN DECK CLASH ROYALE 2025 ===
=====
1. Login
2. Register
3. Keluar
Pilih 1, 2, atau 3:

```

Gambar 4.1 Output Awal

```

Pilih 1, 2, atau 3: 1
Username: Navtaly Juman
Password: 2509106109
Selamat datang ya Navtaly Juman

```

Gambar 4.2 Output Login Admin

```

=== MENU ADMIN ===
1. Lihat Semua Deck
2. Tambah Deck
3. Ubah Deck
4. Hapus Deck
5. Logout
Pilih menu: █

```

Gambar 4.3 Output Menu Admin

```

Pilih menu: 1
ID: 1
Nama Deck: Miner Control
Pemain: Morten
Strategi: Control
Kartu: ['Miner', 'Poison', 'Wall Breaker', 'Skeletons', 'Bats', 'Valkyrie', 'Bomb Tower', 'Log']
ID: 2
Nama Deck: X-Bow 2.9 Cycle
Pemain: Aragon
Strategi: Siege
Kartu: ['X-Bow', 'Tesla', 'Ice Spirit', 'Skeletons', 'Ice Golem', 'Fireball', 'Log', 'Archers']
ID: 3
Nama Deck: LavaLoon
Pemain: Surgical Goblin
Strategi: Air Beatdown
Kartu: ['Lava Hound', 'Balloon', 'Mega Minion', 'Tombstone', 'Minions', 'Fireball', 'Zap', 'Miner']
ID: 4
Nama Deck: Royal Giant Lightning
Pemain: Mohamed Light
Strategi: Beatdown
Kartu: ['Royal Giant', 'Fisherman', 'Lightning', 'Mother Witch', 'Electro Spirit', 'Skeletons', 'Hunter', 'Log']
ID: 5
Nama Deck: Bridge Spam Pekka
Pemain: Ruben
Strategi: Control
Kartu: ['Bandit', 'Battle Ram', 'Royal Ghost', 'Dark Prince', 'Magic Archer', 'Pekka', 'Poison', 'Zap']

```

Gambar 4.4 Output Lihat Semua Deck


```

Pilih menu: 2
Nama Deck: Auto Win
Nama Pemain: No Name
Jenis Strategi: 100% WIN
Masukkan 8 kartu:
Kartu ke-1: Pekka
Kartu ke-2: Fireball
Kartu ke-3: Goblin
Kartu ke-4: Tesla
Kartu ke-5: Miner
Kartu ke-6: Skeletons
Kartu ke-7: Royal Giant
Kartu ke-8: Wizzard
Deck berhasil ditambahkan!

```

Gambar 4.5 Output Tambah Deck

```

Pilih menu: 3
Masukkan ID deck yang ingin diubah: 6
Nama Deck baru: Pasti Kalah
Nama Pemain baru: Malu
Jenis Strategi baru: Nanggis
Masukkan 8 kartu baru:
Kartu ke-1: Goblin
Kartu ke-2: Archer
Kartu ke-3: Mini Pekka
Kartu ke-4: Ballon
Kartu ke-5: Minions
Kartu ke-6: Ice Golem
Kartu ke-7: Log
Kartu ke-8: Skelatons
Deck berhasil diubah!

```

Gambar 4.6 Output Ubah Deck

```

Pilih menu: 4
Masukkan ID deck yang ingin dihapus: 6
Deck berhasil dihapus!

```

Gambar 4.7 Output Hapus Deck

```

Pilih 1, 2, atau 3: 2
Masukkan Username: Mama ku
Masukkan Password: Love you
Akun berhasil dibuat!

```

Gambar 4.8 Output Register User

```

Pilih 1, 2, atau 3: 1
Username: Mama ku
Password: Love you
Selamat datang ya Mama ku
=== MENU USER ===
1. Lihat Semua Deck
2. Logout
Pilih menu:

```

Gambar 4.9 Output Menu User

```

=== MENU USER ===
1. Lihat Semua Deck
2. Logout
Pilih menu: 2
1. Login
2. Register
3. Keluar
Pilih 1, 2, atau 3: 3
Terima kasih sudah menggunakan program ini!

```

Gambar 4.10 Output Selesai

3. Langkah-langkah GIT

5.1 GIT Add

```

PS C:\Users\LENOVO\Documents\praktikum-apd> git add .

```

Gambar 5.1 Git Add

Git Add . ini digunakan untuk menambahkan semua file yang ada di folder saat ini ke staging area.

5.2 GIT Commit

```

PS C:\Users\LENOVO\Documents\praktikum-apd> git commit -m "post-test-apd-7"
[main 5e4ac40] post-test-apd-7
5 files changed, 339 insertions(+), 2 deletions(-)
create mode 100644 kelas/pertemuan 7/pertemuan7.py
create mode 100644 post-test/post-test-apd-6/2509106109-NavtalyJuman-PT-6.pdf
create mode 100644 post-test/post-test-apd-7/2509106109-NavtalyJuman-PT-7.py
create mode 100644 post-test/post-test-apd-7/Flowchart Navtaly.png

```

Gambar 5.2 Git Commit

Git commit -m "post-test-apd-6" ini digunakan untuk menyimpan file yang ada di staging area ke dalam repository lokal, dan -m "post-test-apd-6" untuk menambahkan komentar, pada tahap ini file masih berada di repository lokal komputer belum terupload ke GitHub.

5.3 GIT Push

```
PS C:\Users\LENOVO\Documents\praktikum-apd> git push -u origin main
>>
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 2 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (11/11), 121.77 KiB | 3.93 MiB/s, done.
Total 11 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/naftalijuman/praktikum-apd.git
   54c25de..e8baf00  main -> main
branch 'main' set up to track 'origin/main'.
```

Gambar 5.3 Git Push

Git push ini dipakai untuk mengupload commit dari repository lokal ke repository GitHub di branch main.