UTS Pemrograman PBO



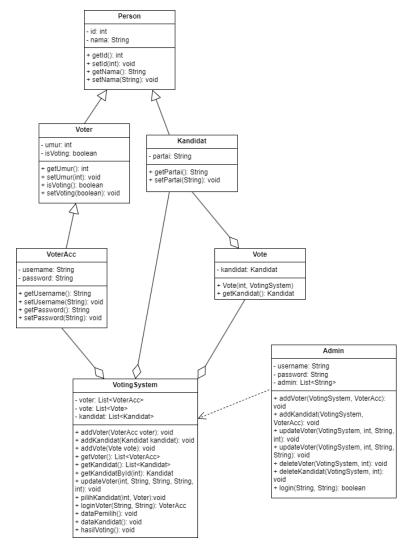
Rafi Ody Prasetyo (2341720180)

D-IV Teknik Informatika Politeknik Negeri Malang Semester 3 2024

1. Deskripsi Project

Sistem Voting adalah plaform yang memungkinkan pengguna untuk melakukan pemungutan suara secara elektronik. Sistem ini dirancang untuk berbagai kebutuhan, seperti pemilihan ketua organisasi, pemilu kampus, survei pendapat, atau polling umum. Sistem ini dibuat untuk menjamin kerahasiaan dan keaslian suara yang diberikan oleh setiap pemilih.

2. UML Diagram



Pada UML di atas terdapat hybrid inheritance yang digunakan untuk mengelola data voter dan kandidat. Class Person merupakan superclass dari inheritance tersebut, dari class Person sub class dapat mengambil data id dan nama. Class Voter memiliki turunan class VoterAcc yang digunakan untuk menambahkan akun. Data dari voter dan kandidat akan tersimpan ke dalam array list yang terdapat pada class VotingSystem. Terdapat class Vote yang digunakan untuk menginputkan pilihan

voter, input pilihan voter akan tersimpan ke dalam array list yang terdapat pada class VotingSystem. Class VotingSystem memiliki relasi kelas yang terhubung dengan class VoterAcc, Kandidat, dan Vote dengan jenis relasi agregasi. Relasi kelas agregasi pada VotingSystem digunakan untuk kebutuhan tipe data array list dan juga beberapa parameter method. Terdapat juga relasi kelas dengan class Admin, yang digunakan untuk mengelola data voter dan kandidat yang terdapat dalam array list. Sehingga relasi yang digunakan antara VotingSystem dan Admin adalah relasi dependency.

3. Script Code

3.1 Person.java

Class ini merupakan superclass yang mewarisi class Voter dan Kandidat. Terdapat setter getter yang digunakan untuk mengontrol nilai variabel dari luar class.

```
public class Person {
    private int id;
    private String nama;

    public Person(int id, String nama) {
        this.id = id;
        this.nama = nama;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getNama() {
        return nama;
    }

    public void setNama(String nama) {
        this.nama = nama;
    }
}
```

3.2 Voter.java

Class ini merupakan subclass yang diwarisi oleh class Person. Pada class ini terdapat atribut isVoting yang digunakan untuk membatasi pemilih agar tidak bisa memilih lebih dari 1 kali. Atribut tersebut tidak masuk ke dalam parameter konstruktor, namun ketika data voter diinputkan secara default isVoting akan bernilai false.

```
public class Voter extends Person{
   private int umur;
   private boolean isVoting;
    public Voter(int id, String nama, int umur) {
        super(id, nama);
        this.umur = umur;
        this.isVoting = false;
    }
    public int getUmur() {
        return umur;
    public void setUmur(int umur) {
        this.umur = umur;
    public boolean isVoting() {
      return isVoting;
    public void setVoting(boolean voting) {
        isVoting = voting;
```

3.3 VoterAcc.java

Class ini merupakan subclass dari class Voter. Class ini dibuat untuk keperluan login ke dalam system voting.

```
public class VoterAcc extends Voter {
    private String username;
    private String password;
    public VoterAcc(int id, String nama, int umur, String
username, String password) {
        super(id, nama, umur);
        this.username = username;
        this.password = password;
    public String getUsername() {
        return username;
    public void setUsername(String username) {
       this.username = username;
    public String getPassword() {
        return password;
    public void setPassword(String password) {
        this.password = password;
```

3.4 Kandidat.java

Kandidat merupakan subclass dari Person.

```
public class Kandidat extends Person{
   private String partai;

   public Kandidat(int id, String nama, String partai) {
        super(id, nama);
        this.partai = partai;
   }

   public String getPartai() {
        return partai;
   }

   public void setPartai(String partai) {
        this.partai = partai;
   }
}
```

3.5 Vote.java

Class ini digunakan untuk input kandidat yang dipilih oleh voter. Terdapat relasi kelas jenis agregasi dengan class Kandidat yang digunakan untuk mengambil data kandidat. Kandidat dipilih berdasarkan id yang ada pada data kandidat. Agar pemilih dapat memilih hanya dengan menginputkan id, class ini dihubungkan dengan class VotingSystem untuk memanggil method getKandidatById. Relasi dengan class VotingSystem adalah relasi jenis dependency.

```
public class Vote {
    private Kandidat kandidat;

    public Vote(int kandidat, VotingSystem votingSystem)
{
        this.kandidat =
    votingSystem.getKandidatById(kandidat);
        if (this.kandidat == null) {
            System.out.println("Kandidat yang anda pilih tidak ada");
        }
    }

    public Kandidat getKandidat() {
        return kandidat;
    }
}
```

3.6 VotingSystem.java

Aplikasi ini menggunakan array list untuk menyimpan data pada aplikasi. Tipe data array list yang digunakan adalah kelas yang ingin disimpan datanya. Untuk menambahkan data ke dalam array list, pada code ini hanya dengan menggunakan fitur yang ada dalam array list yaitu add().

```
import java.util.*;
public class VotingSystem {
   private List<VoterAcc> voter;
   private List<Kandidat> kandidat;
   private List<Vote> vote;
    public VotingSystem() {
       voter = new ArrayList<>();
       kandidat = new ArrayList<>();
       vote = new ArrayList<>();
    public void addVoter(VoterAcc voter) {
        this.voter.add(voter);
    public void addKandidat(Kandidat kandidat) {
       this.kandidat.add(kandidat);
    public void addVote(Vote vote) {
        this.vote.add(vote);
    public List<VoterAcc> getVoter() {
       return voter;
    public List<Kandidat> getKandidat() {
        return kandidat;
```

GetKandidatById() digunakan agar pemilih dapat memilih kandidat hanya berdasarkan id saja. Kemudian method updateVoter() digunakan untuk mengupdate data dari pemilih berdasarkan akun yang digunakan login pada system, sehingga data yang muncul hanya data akun yang sedang login. Method pilihKandidat() digunakan untuk menginputkan pilihan. Pada method tersebut terdapat percabang yang apabila atribut isVoting bernilai false maka pemilih masih dapat memilih. Hal ini dilakukan agar pemilih tidak bisa memilih lebih dari 1 kali.

```
public Kandidat getKandidatById(int id) {
        for (Kandidat kandidat : kandidat) {
            if (kandidat.getId() == id) {
                return kandidat;
        return null;
    }
    public void updateVoter (int idVoter, String newName,
String newUsername, String newPassword, int newAge) {
        for (VoterAcc voter : getVoter()) {
                voter.setNama(newName);
                voter.setUsername(newUsername);
                voter.setPassword(newPassword);
                voter.setUmur(newAge);
    }
    public void pilihKandidat(int idKandidat, Voter
voter) {
        Kandidat kandidat = getKandidatById(idKandidat);
        if (kandidat != null) {
            if (!voter.isVoting()) {
                Vote vote = new Vote(idKandidat, this);
                addVote(vote);
                System.out.println("Anda telah memilih "
+ kandidat.getNama() + " dengan partai " +
kandidat.getPartai());
                voter.setVoting(true);
            } else {
                System.out.println("Anda sudah melakukan
voting!");
        } else {
            System.out.println("Kandidat tidak
ditemukan!");
```

Method loginVoter() digunakan untuk login pemilih. Pada method tersebut program akan menyamakan inputan username dan password dengan data yang ada pada arraylist. Untuk method dataPemilih() dan dataKandidat() digunakan untuk menampilkan data dari pemilih dan kandidat.

```
public VoterAcc loginVoter (String username, String
password) {
      for (VoterAcc voterAcc : voter) {
        if (voterAcc.getUsername().equals(username)
&& voterAcc.getPassword().equals(password)) {
         return voterAcc;
      }
     return null;
  public void dataPemilih() {
   System.out.println("+---+
+----+");
 System.out.println("| ID | Nama
Umur | Status Memilih |");
    System.out.println("+---+
 ----+");
     for (Voter voter2 : voter) {
         System.out.printf("| %-2d | %-20s | %-5d | %-
15s |\n",
        voter2.getId(),
         voter2.getNama(),
         voter2.getUmur(),
         (voter2.isVoting() ? "sudah memilih" : "belum
memilih"));
     System.out.println("+---+
  ----+");
     System.out.println("Jumlah Pemilih: " +
voter.size());
 }
  public void dataKandidat() {
   System.out.println("+---+
 ------;
 System.out.println("| ID | Nama
       |");
    System.out.println("+---+
+----+");
     for (Kandidat kandidat : kandidat) {
       System.out.printf("| %-2d | %-20s | %-16s
| n'',
            kandidat.getId(),
            kandidat.getNama(),
            kandidat.getPartai());
     System.out.println("+---+
+----+");
 }
```

Method hasilVoting() digunakan untuk menghitung jumlah suara yang masuk. Pada method tersebut ditentukan pula pemenang berdasarkan suara terbesar.

```
public void hasilVoting() {
      int totalVote = 0;
      Kandidat pemenang = null;
      int maxVote = 0;
      System.out.println("+---+
      System.out.println("| ID | Nama Kandidat
| Partai | Jumlah Suara |");
      System.out.println("+---+
      ----+");
      for (Kandidat kandidat : kandidat) {
          int kandidatTerpilih = 0;
          for (Vote vote : vote) {
             if (vote.getKandidat().equals(kandidat))
                 kandidatTerpilih++;
             }
          }
          totalVote += kandidatTerpilih;
          if (kandidatTerpilih > maxVote) {
             maxVote = kandidatTerpilih;
             pemenang = kandidat;
          System.out.printf("| %-2d | %-34s | %-18s |
%-12d |\n", kandidat.getId(), kandidat.getNama(),
kandidat.getPartai(), kandidatTerpilih);
          System.out.println("+---+
      }
      System.out.println("Total Suara: " + totalVote);
      if (pemenang == null) {
          System.out.println("Suara masih kosong!");
          System.out.println("Pemenang: " +
pemenang.getNama() + " (" + pemenang.getPartai() + ")
dengan " + maxVote + " suara");
}
```

3.7 Admin.java

Class ini merupakan class yang digunakan untuk mengelola data voter dan kandidat. CRUD pada class ini menggunakan fitur yang ada pada library array list. Class ini memiliki relasi dependency dengan class VotingSystem karena hanya memanggil class tersebut pada method yang memerlukan pemanggilan arraylist.

```
import java.util.*;
public class Admin {
    private String username;
   private String password;
   private String login;
   ArrayList<String> admin = new ArrayList<String>();
   public Admin(String username, String password) {
        this.login = username + password;
        admin.add(username);
        admin.add(password);
    public void addVoter (VotingSystem votingSystem,
VoterAcc voter) {
        votingSystem.addVoter(voter);
   public void addKandidat(VotingSystem votingSystem,
Kandidat kandidat) {
       votingSystem.addKandidat(kandidat);
    public void updateVoter(VotingSystem votingSystem, int
voterId, String newName, int newAge) {
        for (Voter voter : votingSystem.getVoter()) {
            if (voter.getId() == voterId) {
                voter.setNama(newName);
                voter.setUmur(newAge);
                break;
            } else {
                System.out.println("Data tidak
ditemukan!");
    }
```

```
public void updateKandidat(VotingSystem votingSystem, int
idKandidat, String newName, String newPartai) {
        for (Kandidat kandidat : votingSystem.getKandidat()) {
            if (kandidat.getId() == idKandidat) {
                kandidat.setNama(newName);
                kandidat.setPartai(newPartai);
            } else {
                System.out.println("Data tidak ditemukan!");
        }
    public void deleteVoter (VotingSystem votingSystem, int
voterId) {
        List<VoterAcc> voterList = votingSystem.getVoter();
        boolean vtr = voterList.stream().anyMatch(v -> v.getId()
== voterId);
        if (vtr) {
            voterList.removeIf(voter -> voter.getId() ==
voterId);
            System.out.println("Pemilih berhasil dihapus!");
        } else {
            System.out.println("Pemilih tidak ditemukan!");
    public void deleteKandidat(VotingSystem votingSystem, int
kandidatId) {
        List<Kandidat> kandidatList =
votingSystem.getKandidat();
        boolean kandidat = kandidatList.stream().anyMatch(k ->
k.getId() == kandidatId);
        if (kandidat) {
            kandidatList.removeIf(voter -> voter.getId() ==
kandidatId);
            System.out.println("Kandidat berhasil dihapus!");
        } else {
            System.out.println("Kandidat tidak ditemukan!");
    }
    public boolean login(String username, String password) {
        String auth = username + password;
        return this.login.equals(auth);
```

4. Keluaran

Tampilan login:

```
Selamat Datang di Aplikasi Sistem Voting
Username: admin
Password: admin123
```

Menu admin:

```
Admin Menu:
1. Tambah Pemilih
2. Daftar Pemilih
3. Tambah Kandidat
4. Daftar Kandidat
5. Update Voter
6. Update Kandidat
7. Delete Voter
8. Delete Kandidat
9. Hasil Voting
10. Logout
0. Exit
```

Tambah pemilih:

```
Enter your pilih: 1
Masukkan ID: 2
Masukkan nama: Budi
Masukkan username: budi
Masukkan password: budi123
Masukkan umur: 19
```

Daftar pemilih:

ID Nama	Umur	++ Status Memilih
1 dave 2 Budi	21 19	belum memilih belum memilih
Jumlah Pemilih: 2	+	++

Tambah kandidat:

```
Masukkan ID: 3
Masukkan nama: Soeharto
Masukkan partai: Golkar
```

Daftar kandidat:

<u> </u>	, oui pririii .	·
ID	Nama	Partai
2	Prabowo Jokowi Soeharto	Gerindra PDIP Golkar

Update voter:

1 1	Umur	
1	21 19	belum memilih belum memilih
Jumlah Pemilih: 2 Masukkan id voter untuk updat Masukkan nama baru: Doni Masukkan umur baru: 23		•

		 Status Memilih
1	dave	belum memilih belum memilih

Update kandidat:

ID	Nama	+ Partai	
! - !	Prabowo Jokowi Soeharto	Gerindra PDIP Golkar	
Hasukkan id kandidat untuk update: 2 Masukkan nama baru: Mulyono Masukkan partai baru: PKB			

++		-
ID	Nama	Partai
++		
1	Prabowo	Gerindra
2	Mulyono	PKB
3	Soeharto	Golkar
++		++

Delete voter:

++	+ Umur	 Status Memilih
++	+	
1 dave	21	belum memilih
2 Doni	23	belum memilih
Jumlah Pemilih: 2 Masukkan ID pemilih: 2 Pemilih berhasil dihapus!	+	· -

	_4	
ID Nama	Umur	Status Memilih
1 dave	21	belum memilih
Jumlah Pemilih: 1	-+	+

Delete kandidat:



Hasil voting:

ID Nama Kandidat	H Partai	tt Jumlah Suara
1 Prabowo	Gerindra	0
2 Mulyono	PKB	0
Total Suara: 0 Suara masih kosong!	•	***************************************

Tampilan menu pemilih:

Aplikasi Sistem Voting
Selamat Datang dave
1. Pilih Kandidat
2. Hasil Voting
3. Update Akun
4. Logout
0. Exit
Pilih:

Pilih kandidat:

Daftar Kandidat: 1. Prabowo (Gerindra) 2. Mulyono (PKB) Masukkan ID kandidat: 1 Anda telah memilih Prabowo dengan partai Gerindra

Hasil voting:

++	+	+	
ID Nama Kandidat	Partai	Jumlah Suara	
1 Prabowo	Gerindra	1	
2 Mulyono	PKB	0	
++			

Update akun:

Pilih:

Nama: dave Username: dave Password: dave123 Umur: 21 Apakah anda yakin ingin mengganti data akun? (y/n) Pilih: y Masukkan nama: gusta Masukkan username: gusta123 Masukkan password: gusta123 Masukkan umur: 22 Aplikasi Sistem Voting Selamat Datang gusta 1. Pilih Kandidat 2. Hasil Voting 3. Update Akun 4. Logout 0. Exit