

Aplikasi Pendaftaran Mahasiswa di Perguruan Tinggi





Kelompok 2



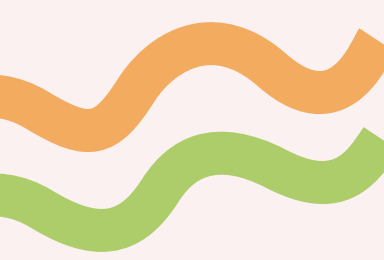
Sherly Kartika Dewi Azahara
103062300007



**Disya Nabila Shativa
Resnanda**
103062330024



Hawa Sakura
103062300011



Pengantar

LATAR BELAKANG MASALAH

- Pertumbuhan perguruan tinggi yang pesat memerlukan sistem pendaftaran yang efisien
- Banyak institusi masih menggunakan proses manual yang rentan terhadap kesalahan dan tidak efisien
- Di era modern, banyak perguruan tinggi beralih ke sistem digital untuk mengelola pendaftaran mahasiswa




Pengantar

MOTIVASI

- Digitalisasi membantu dalam pengelolaan data yang lebih efisien dan akurat
- Mengelola data secara elektronik meningkatkan efisiensi operasional, mengurangi kesalahan manusia, dan mempermudah akses informasi bagi calon mahasiswa dan staf administrasi
- Aplikasi ini dirancang untuk meningkatkan efisiensi dalam pengolahan data pendaftaran dan memastikan akurasi data, untuk pengambilan keputusan dalam proses penerimaan mahasiswa



Landasan Teori



1. Prosedur : kumpulan algoritma/instruksi terhadap suatu proses tertentu yang menghasilkan suatu akibat langsung

2. Array : kumpulan elemen dengan tipe data yang sama yang memiliki ukuran tetap
Tipe Bentukan : (struct) kumpulan dari beberapa tipe data yang berbeda yang didefinisikan oleh pengguna

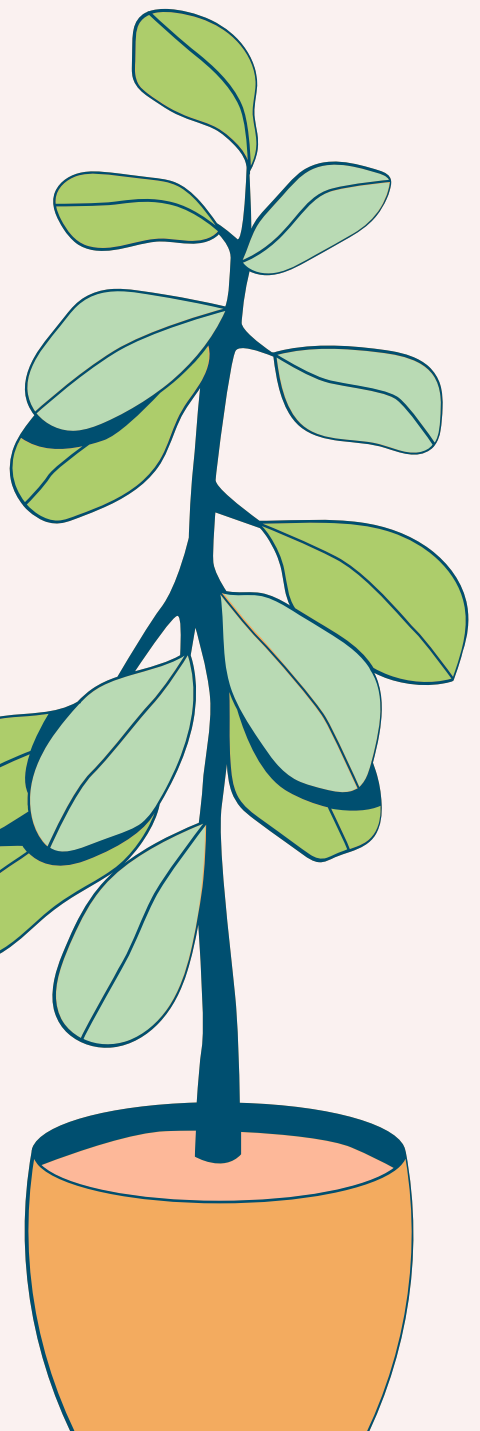
3. Algoritma Searching : untuk menemukan nilai maksimum atau minimum pada kumpulan data

- Sequential Search (data acak)
- Binary Search (data terurut)

4. Algoritma Sorting : proses untuk mengatur di mana posisi data yang seharusnya

- Selection Sort : mencari posisi/indeks maksimum/minimum pada array, lalu ditukar
- Insertion Sort : mencari tempat yang “tepat” untuk setiap nilai kemudian menyisipkan pada tempat yang seharusnya

5. Function : blok kode yang menerima argumen dan mengembalikan nilai



Pembahasan

Aplikasi Pendaftaran Mahasiswa di Perguruan Tinggi ini memfasilitasi pengelolaan data pendaftaran secara efisien dan terstruktur, meliputi data calon mahasiswa dan data jurusan. Pengguna aplikasi, termasuk petugas admin dan calon mahasiswa, dapat menambah, mengubah, dan menghapus data. Aplikasi ini juga memungkinkan pengguna untuk melihat data pendaftaran berdasarkan jurusan, serta status penerimaan atau penolakan mahasiswa. Selain itu, manajemen nilai tes calon mahasiswa diatur sedemikian rupa sehingga dapat menentukan hasil seleksi dengan tepat. Fitur lainnya mencakup pengurutan data mahasiswa berdasarkan nilai tes, jurusan, atau nama, yang membantu dalam proses analisis dan pengambilan keputusan. Dengan antarmuka yang ramah pengguna dan sistem keamanan data yang kuat, aplikasi ini meningkatkan efisiensi dan akurasi dalam proses pendaftaran di perguruan tinggi.

Algoritma Aplikasi Pendaftaran Mahasiswa

MENU UTAMA

1. Register Mahasiswa Baru

1.1. Tampilkan form registrasi dengan input berikut :

- Email
- Nama Lengkap
- Jurusan
- Nomor Telepon

1.2. Validasi input:

- Email: Cek apakah email sudah terdaftar di sistem. Jika sudah, tampilkan pesan error dan minta pengguna memasukkan email yang berbeda.
- Nama Lengkap, Jurusan, Nomor Telepon: Pastikan semua field diisi.

1.3. Simpan data mahasiswa baru ke dalam database dengan struktur data seperti: {email, nama lengkap, jurusan, nomor telepon}.

1.4. Konfirmasi pendaftaran berhasil dan tampilkan pesan sukses.

2. Cek Nilai Tes dan Status Kelulusan

2.1. Tampilkan form input untuk email.

2.2. Input email dari pengguna.

2.3. Cari data mahasiswa berdasarkan email yang diberikan.

- Jika data tidak ditemukan, tampilkan pesan error.

2.4. Tampilkan nilai tes dan status kelulusan (Diterima/Tidak Diterima) untuk email tersebut

Algoritma Aplikasi Pendaftaran Mahasiswa

3. Masuk Sebagai Admin

- Tampilkan form input untuk password admin.
- Input password dari pengguna.
- Validasi password:

Jika password adalah "AllAdmin555", izinkan akses ke menu admin. Jika salah, tampilkan pesan error.

MENU ADMIN

3.1. Kelola Data Jurusan

3.1.1. Tambah Jurusan

- Tampilkan form input untuk nama jurusan baru.
- Input nama jurusan dari pengguna.
- Validasi nama jurusan:
- Pastikan nama jurusan tidak kosong dan tidak duplikat.
- Tambahkan nama jurusan ke dalam database jurusan.
- Konfirmasi bahwa jurusan berhasil ditambahkan.

3.1.2. Ubah Jurusan

- Tampilkan daftar jurusan yang ada.
- Pilih jurusan yang akan diubah.
- Tampilkan form untuk mengubah nama jurusan.
- Input nama jurusan yang baru dari pengguna.
- Validasi perubahan:
- Pastikan nama jurusan baru tidak kosong dan tidak duplikat.
- Perbarui nama jurusan dalam database.
- Konfirmasi bahwa jurusan berhasil diubah.

Algoritma Aplikasi Pendaftaran Mahasiswa

3.1.3. Hapus Jurusan

- Tampilkan daftar jurusan yang ada.
- Pilih jurusan yang akan dihapus.
- Hapus jurusan dari database.
- Konfirmasi bahwa jurusan berhasil dihapus.

3.2. Kelola Nilai Tes Mahasiswa

Mengelola data nilai tes mahasiswa.

3.2.1. Tambah Nilai Tes

- Tampilkan daftar mahasiswa tanpa nilai tes.
- Pilih mahasiswa untuk ditambahkan nilai tes.
- Input nilai tes dari pengguna.
- Validasi nilai tes: Pastikan nilai berada dalam rentang yang valid.
- Simpan nilai tes ke dalam database untuk mahasiswa yang dipilih.
- Konfirmasi bahwa nilai tes berhasil ditambahkan.

3.2.2. Ubah Nilai Tes

- Tampilkan daftar mahasiswa dengan nilai tes.
- Pilih mahasiswa yang nilainya akan diubah.
- Tampilkan form untuk mengubah nilai tes.
- Input nilai tes yang baru dari pengguna.
- Validasi nilai tes yang baru.
- Perbarui nilai tes dalam database.
- Konfirmasi bahwa nilai tes berhasil diubah.

3.2.3. Hapus Nilai Tes

- Tampilkan daftar mahasiswa dengan nilai tes.
- Pilih mahasiswa yang nilainya akan dihapus.
- Hapus nilai tes dari database.
- Konfirmasi bahwa nilai tes berhasil dihapus.

3.3. Tampilkan Data Mahasiswa

3.3.1. Yang Mendaftar di Jurusan Tertentu

- Tampilkan daftar jurusan.
- Pilih jurusan untuk melihat mahasiswa yang mendaftar.
- Tampilkan daftar mahasiswa yang mendaftar di jurusan tersebut.

3.3.2. Berdasarkan Status Kelulusan

- Tampilkan opsi untuk status kelulusan: "Diterima" atau "Ditolak".
- Pilih status kelulusan yang diinginkan.
- Tampilkan daftar mahasiswa dengan status kelulusan yang dipilih.

3.3.3. Sorted by Nilai Tes

- Dapatkan daftar semua mahasiswa dengan nilai tes.
- Urutkan daftar berdasarkan nilai tes secara descending (dari yang tertinggi ke terendah).
- Tampilkan daftar mahasiswa yang sudah diurutkan.

Algoritma Aplikasi Pendaftaran Mahasiswa

3.3.4. Sorted by Jurusan

- Dapatkan daftar semua mahasiswa.
- Urutkan daftar berdasarkan nama jurusan.
- Tampilkan daftar mahasiswa yang sudah diurutkan.

3.3.5. Sorted by Huruf Depan Nama

- Dapatkan daftar semua mahasiswa.
- Urutkan daftar berdasarkan huruf depan nama mahasiswa.
- Tampilkan daftar mahasiswa yang sudah diurutkan.

3.4. Kelola Data Mahasiswa

3.4.1. Edit Data Mahasiswa

- Tampilkan daftar mahasiswa.
- Pilih mahasiswa yang datanya akan diubah.
- Tampilkan form untuk mengubah data mahasiswa (nama, jurusan, nomor telepon).
- Input perubahan data dari pengguna.
- Validasi perubahan data.
- Perbarui data mahasiswa dalam database.
- Konfirmasi bahwa data mahasiswa berhasil diubah.

3.4.2. Hapus Data Mahasiswa

- Tampilkan daftar mahasiswa.
- Pilih mahasiswa yang datanya akan dihapus.
- Hapus data mahasiswa dari database.
- Konfirmasi bahwa data mahasiswa berhasil dihapus.

Pseudocode - Tipe Bentuk dan Array

// Tipe Bentuk Mahasiswa

Tipe Mahasiswa

Email: String

Nama: String

Jurusan: String

NoTelepon: String

NilaiTes: Float

Status: String

// Tipe Bentuk Jurusan

Tipe Jurusan

Nama: String

// Deklarasi variabel global

Array of Mahasiswa mahasiswaList

// Menyimpan data mahasiswa

Array of Jurusan jurusanList

// Menyimpan data jurusan yang tersedia

String adminPassword = "AllAdmin555" // Password
untuk akses admin

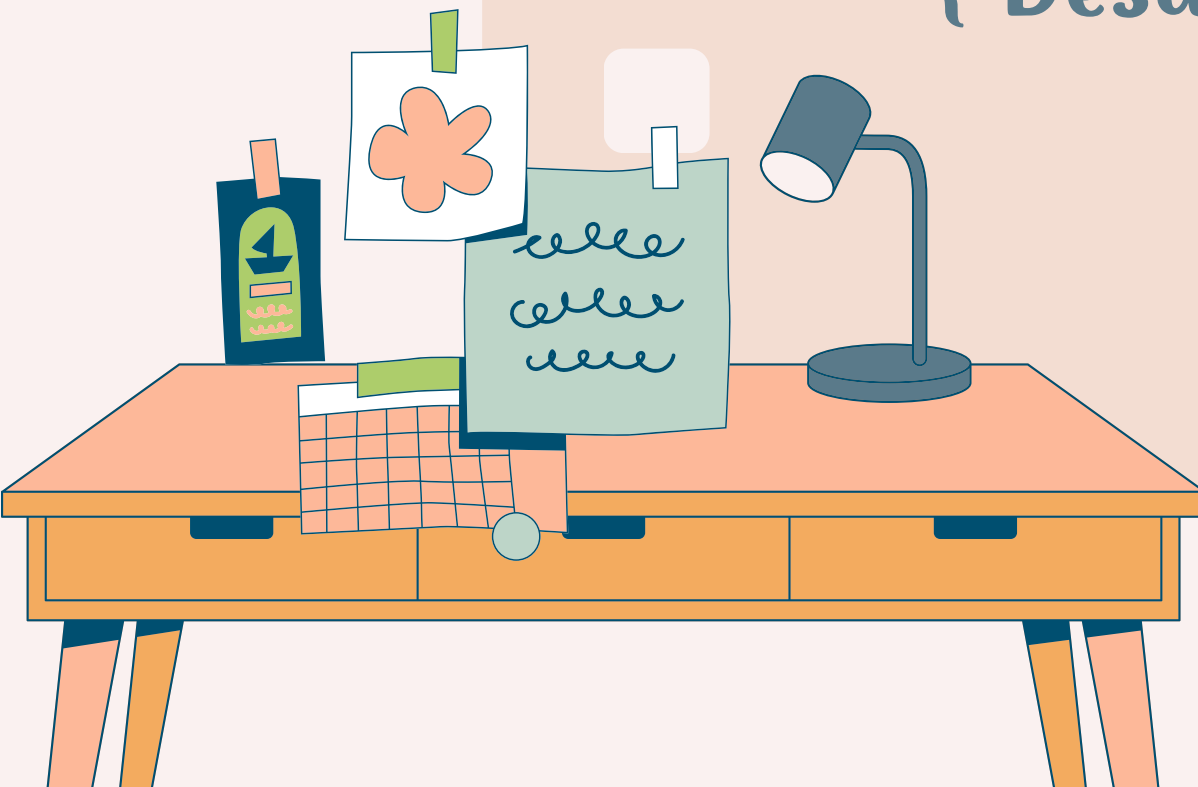
// Isi dari jurusanList jurusanList = [
{"Informatika"}, {"Teknologi Informasi"},
{"Sistem Informasi"}, {"Teknik
Telekomunikasi"}, {"Teknik Elektro"},
{"Teknik Industri"}, {"Desain Komunikasi
Visual"}]



Pseudocode - Algoritma Searching

Pencarian Email Mahasiswa

```
// Isi dari jurusanList jurusanList = [ {"Informatika"},  
{"Teknologi Informasi"}, {"Sistem Informasi"}, {"Teknik  
Telekomunikasi"}, {"Teknik Elektro"}, {"Teknik Industri"},  
{"Desain Komunikasi Visual"} ]
```



Pseudocode - Algoritma Selection Sort

// Fungsi untuk mengurutkan mahasiswa berdasarkan NilaiTes menggunakan Selection Sort

Fungsi sortByNilaiTes(ascending: Boolean)

Integer n = Panjang dari mahasiswaList

Untuk i dari 0 sampai n-1

Integer idx_min = i

Untuk j dari i+1 sampai n

Jika ascending dan mahasiswaList[j].NilaiTes < mahasiswaList[idx_min].NilaiTes

idx_min = j

Jika tidak ascending dan mahasiswaList[j].NilaiTes > mahasiswaList[idx_min].NilaiTes

idx_min = j

// Tukar mahasiswaList[i] dengan mahasiswaList[idx_min]

Mahasiswa temp = mahasiswaList[i]

mahasiswaList[i] = mahasiswaList[idx_min]

mahasiswaList[idx_min] = temp



Pseudocode - Algoritma Insertion Sort

// Fungsi untuk mengurutkan mahasiswa berdasarkan Nama menggunakan Insertion Sort

Fungsi sortByNama(ascending: Boolean)

Integer n = Panjang dari mahasiswaList

Untuk i dari 1 sampai n-1

Mahasiswa key = mahasiswaList[i]

Integer j = i - 1

Jika ascending

// Pindahkan elemen yang lebih besar dari key ke satu posisi ke depan

Sementara j >= 0 dan mahasiswaList[j].Nama > key.Nama

 mahasiswaList[j + 1] = mahasiswaList[j]

 j = j - 1

Jika tidak ascending

// Pindahkan elemen yang lebih kecil dari key ke satu posisi ke depan

Sementara j >= 0 dan mahasiswaList[j].Nama < key.Nama

 mahasiswaList[j + 1] = mahasiswaList[j]

 j = j - 1

mahasiswaList[j + 1] = key



Kesimpulan

Aplikasi Pendaftaran Mahasiswa di Perguruan Tinggi ini berhasil mengotomatisasi dan menyederhanakan proses pendaftaran yang sebelumnya manual dan memakan waktu. Dengan mengelola data calon mahasiswa dan jurusan secara elektronik, aplikasi ini memberikan kemudahan bagi petugas admin untuk melakukan penambahan, pengubahan, dan penghapusan data. Fitur penyajian data berdasarkan jurusan dan status penerimaan memberikan transparansi dan memudahkan pemantauan pendaftaran. Pengelolaan nilai tes yang terintegrasi memungkinkan evaluasi yang lebih objektif dan efisien dalam menentukan kelayakan mahasiswa. Selain itu, kemampuan aplikasi untuk mengurutkan data berdasarkan nilai tes, jurusan, dan nama mahasiswa, membantu perguruan tinggi dalam mengatur dan menganalisis data pendaftaran dengan lebih baik. Aplikasi ini tidak hanya meningkatkan efisiensi operasional tetapi juga memastikan integritas dan keamanan data, sehingga dapat diandalkan dalam proses penerimaan mahasiswa. Keberhasilan implementasi aplikasi ini menunjukkan potensi besar dalam transformasi digital proses administrasi di perguruan tinggi, mendorong kemajuan dan efisiensi lebih lanjut dalam pengelolaan pendidikan tinggi.

**Terima
Kasih**

