**LAPORAN PRAKTIK**

**STRUKTUR DATA**

**Dosen Pengampu :**

Bonita Destiana S.Pd., M.Pd.



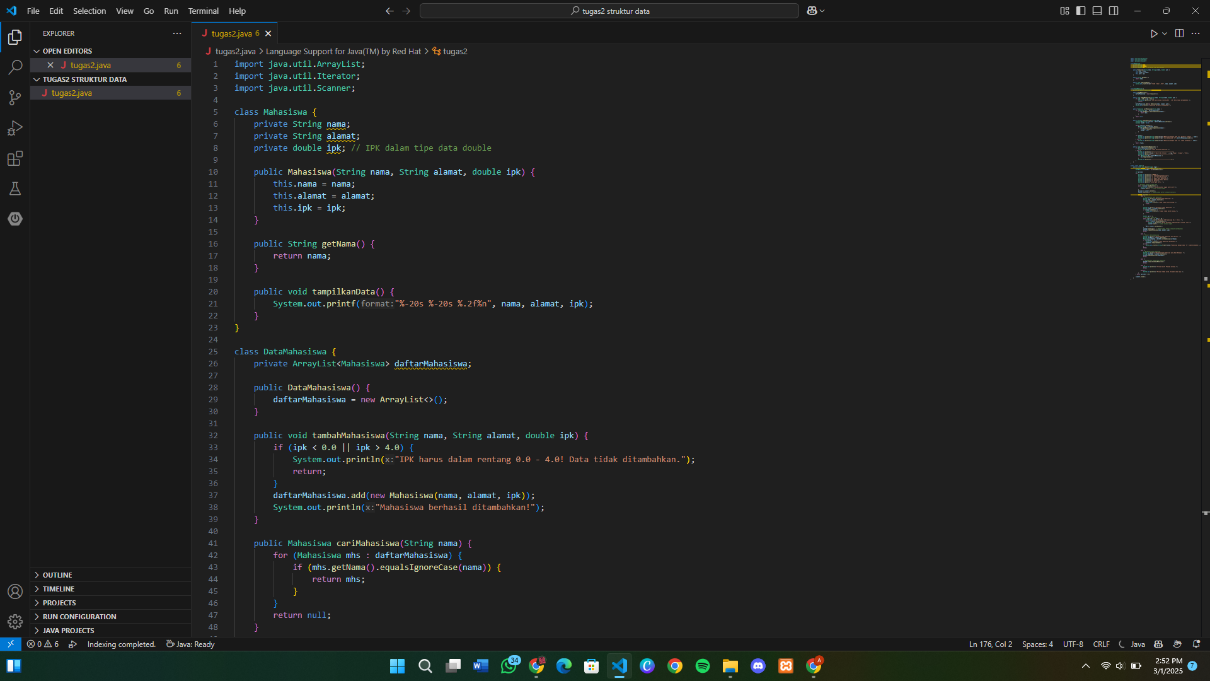
**Dibuat Oleh:**

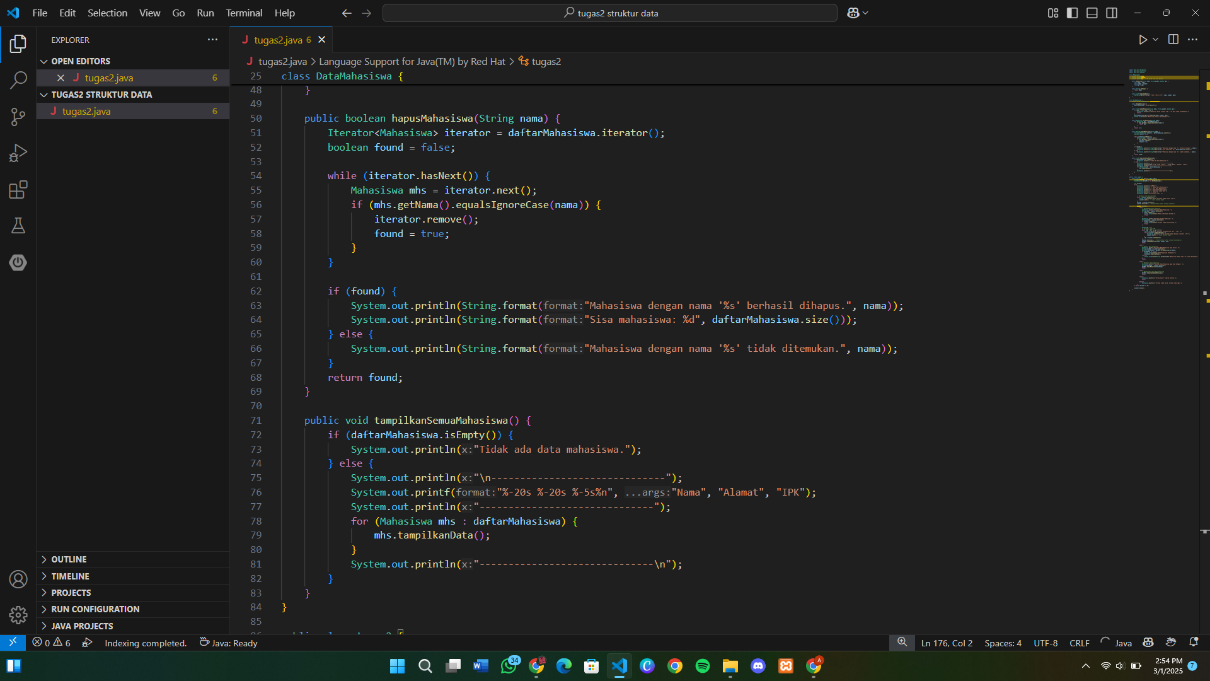
Alfan Ghulam Haydar / 24050530023

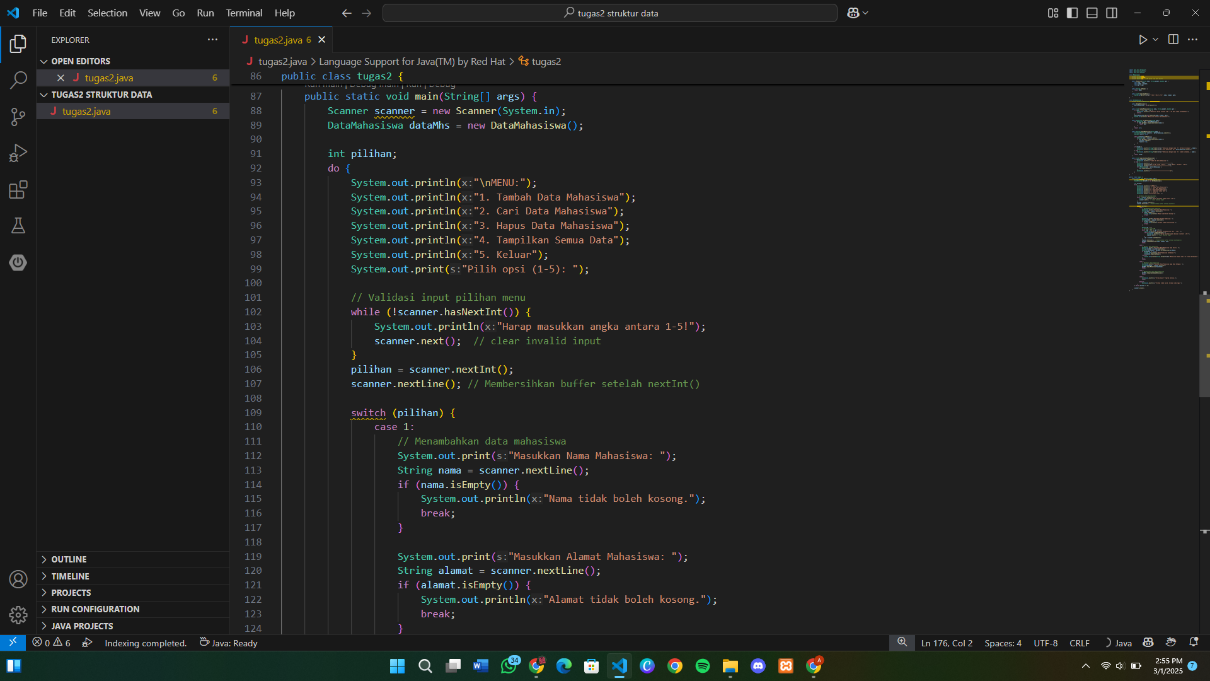
PRODI PENDIDIKAN TEKNIK INFORMATIKA

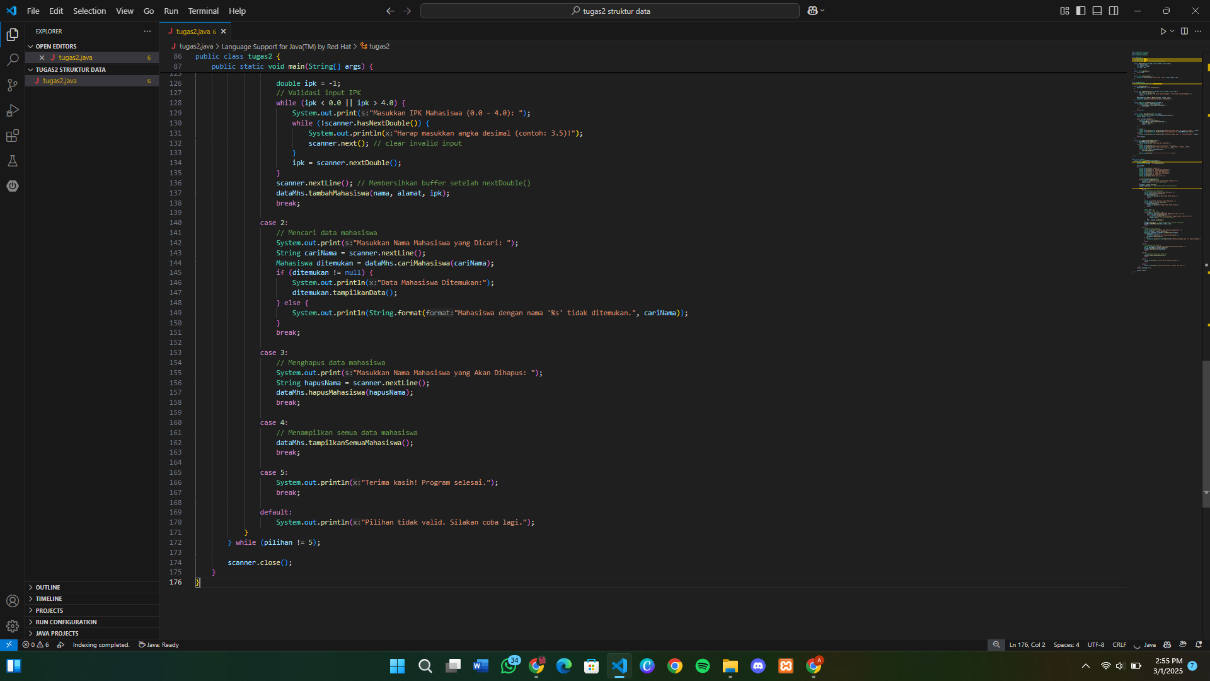
UNIVERSITAS NEGERI YOGYAKARTA

program









|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Nama fungsi | Syntax | Penjelasan | Hasil Running |
| 1. | TambahMahasiswa() | Public void tambahMahasiswa(String nama, String alamat, double ipk) {  if (ipk < 0.0 || ipk > 4.0) {  System.out.println("IPK harus dalam rentang 0.0 - 4.0! Data tidak ditambahkan.");  return;  }  daftarMahasiswa.add(new Mahasiswa(nama, alamat, ipk));  System.out.println("Mahasiswa berhasil ditambahkan!");  } |  |  |
| 2. | cariMahasiswa() | public Mahasiswa cariMahasiswa(String nama) {  for (Mahasiswa mhs : daftarMahasiswa) {  if (mhs.getNama().equalsIgnoreCase(nama)) {  return mhs;  }  }  return null;  } | Mencari mahasiswa berdasarkan nama, Menggunakan equalsIgnoreCase() untuk membandingkan nama tanpa memperhatikan huruf besar/kecil, Jika mahasiswa ditemukan, akan mengembalikan objek Mahasiswa, Jika tidak ditemukan, mengembalikan null. |  |
| 3. | hapuMahasiswa() | public boolean hapusMahasiswa(String nama) {  Iterator<Mahasiswa> iterator = daftarMahasiswa.iterator();  boolean found = false;  while (iterator.hasNext()) {  Mahasiswa mhs = iterator.next();  if (mhs.getNama().equalsIgnoreCase(nama)) {  iterator.remove();  found = true;  }  }  if (found) {  System.out.println(String.format("Mahasiswa dengan nama '%s' berhasil dihapus.", nama));  System.out.println(String.format("Sisa mahasiswa: %d", daftarMahasiswa.size()));  } else {  System.out.println(String.format("Mahasiswa dengan nama '%s' tidak ditemukan.", nama));  }  return found;  } | Menggunakan Iterator untuk menghapus mahasiswa agar menghindari ConcurrentModificationException,  Jika nama mahasiswa ditemukan, maka akan dihapus dari daftar,  Menampilkan pesan sukses atau gagal jika tidak ditemukan. |  |
| 4. | tampilkanSemuaMahasiswa() | public void tampilkanSemuaMahasiswa() {  if (daftarMahasiswa.isEmpty()) {  System.out.println("Tidak ada data mahasiswa.");  } else {  System.out.println("\n------------------------------");  System.out.printf("%-20s %-20s %-5s%n", "Nama", "Alamat", "IPK");  System.out.println("------------------------------");  for (Mahasiswa mhs : daftarMahasiswa) {  mhs.tampilkanData();  }  System.out.println("------------------------------\n");  }  } | Menampilkan tabel mahasiswa dengan format rapi, Jika tidak ada mahasiswa, menampilkan pesan "Tidak ada data mahasiswa.",  Menggunakan loop for untuk mencetak setiap mahasiswa dalam daftar. |  |
| 5. | Keluar() | case 5:  System.out.println("Terima kasih! Program selesai.");  break; | Jika pengguna memilih opsi 5, maka program akan menampilkan pesan perpisahan,  break; akan menghentikan perulangan do-while, sehingga program keluar. |  |
| 6. | Menu() | do {  System.out.println("\nMENU:");  System.out.println("1. Tambah Data Mahasiswa");  System.out.println("2. Cari Data Mahasiswa");  System.out.println("3. Hapus Data Mahasiswa");  System.out.println("4. Tampilkan Semua Data");  System.out.println("5. Keluar");  System.out.print("Pilih opsi (1-5): ");  // Validasi input pilihan menu  while (!scanner.hasNextInt()) {  System.out.println("Harap masukkan angka antara 1-5!");  scanner.next(); // clear invalid input  }  pilihan = scanner.nextInt();  scanner.nextLine(); // Membersihkan buffer setelah nextInt()  switch (pilihan) {  case 1:  // Menambahkan data mahasiswa  // ... (kode untuk menambah mahasiswa)  break;  case 2:  // Mencari data mahasiswa  // ... (kode untuk mencari mahasiswa)  break;  case 3:  // Menghapus data mahasiswa  // ... (kode untuk menghapus mahasiswa)  break;  case 4:  // Menampilkan semua data mahasiswa  // ... (kode untuk menampilkan semua mahasiswa)  break;  case 5:  System.out.println("Terima kasih! Program selesai.");  break;  default:  System.out.println("Pilihan tidak valid. Silakan coba lagi.");  }  } while (pilihan != 5);  scanner.close(); | Program menggunakan loop do-while untuk menampilkan menu hingga pengguna memilih keluar (5), dengan validasi input agar hanya angka 1-5 yang diterima. Switch case menangani opsi: 1 tambah data, 2 cari mahasiswa, 3 hapus mahasiswa, 4 tampilkan semua data, dan 5 keluar. Default case menangani input tidak valid. |  |