**ULANGAN AKHIR SEMESTER**

Dosen : I Made Gede Sri Artha

Matkul : Object Oriented Programming

Hari/Tgl : Selasa, 01 July 2025

Nama : Gabriel Friskila Dinzie S

NIM : 2301010429

Kelas : Malam – J

**DOKUMENTASI DESAIN APLIKASI**

**PENCATATAN GAJI KANTOR SKETCHI STUDIO**

1. **Project:** Aplikasi Pencatatan Gaji Kantor Sketchi Studio
2. **Deskripsi**
3. **Deskripsi Project**

Aplikasi Pencatatan Gaji di kantor Sketchi Studio merupakan aplikasi yang dibuat dengan bahasa pemrograman **Java** menggunakan **Swing** sebagai GUI dan **MySQL** sebagai basis data yang dirancang untuk membantuk karyawan Sketchi Studio dalam mencatat gaji mereka masing-masing secara efisien dan menjumlahkan total gaji secara otomatis, tanpa perlu menghitung secara manual.

Dalam kantor saya bekerja yaitu Sketchi Studio untuk memasukkan data dan menghitung gaji masih menggunakan excel dan dihitung menggunakan rumus “SUM”, maka dari itu saya membuat aplikasi pencatatan gaji ini, agar lebih cepat dan efisien.

1. **Teknik/ Method**
2. **Acces Modifier**

* **Penjelasan**: Mengatur tingkat akses suatu class, method, atau variabel (mis. public, private).
* **Alasan Penggunaan**: Untuk membatasi akses agar method internal tidak bisa dipanggil sembarangan dari luar kelas. Menjaga keamanan dan modularitas program.

1. **Constructor**

* **Penjelasan**: Method khusus yang dipanggil saat objek dibuat, digunakan untuk inisialisasi awal.
* **Alasan Penggunaan**: Mengatur tampilan awal aplikasi, menginisialisasi komponen, model tabel, dan data ketika form dibuka.

1. **Encapsulation**

* **Penjelasan**: Menyembunyikan detail implementasi dalam method dan hanya mengekspos bagian yang perlu.
* **Alasan Penggunaan**: Agar kode lebih terstruktur dan tidak saling mengganggu antarbagian. Mempermudah pemeliharaan dan pembaruan program.

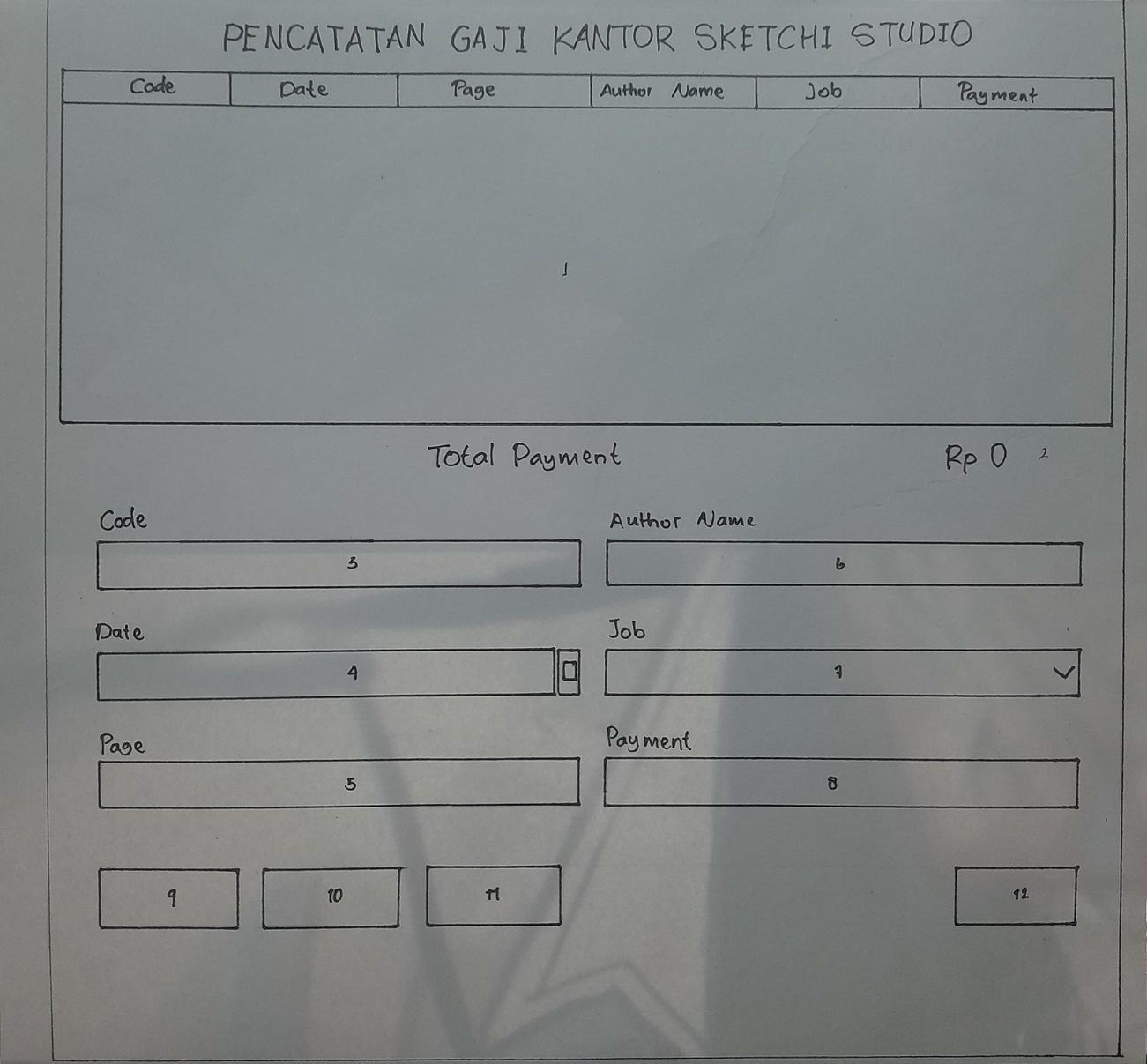
1. **Method Inheritance**

* **Penjelasan**: Pewarisan method dari superclass (dalam JFrame dan komponen Swing lainnya).
* **Alasan Penggunaan**: Memungkinkan untuk menggunakan fungsi bawaan dari Java Swing tanpa harus menulis ulang, seperti tampilan GUI dan pengendalian event.

1. **Polymorphism**

* **Penjelasan**: Kemampuan method untuk menerima objek berbeda yang masih dalam satu hirarki class/interface.
* **Alasan Penggunaan**: Memudahkan manipulasi komponen GUI seperti ComboBox dan TableModel yang bisa diatur menggunakan berbagai jenis objek model.

1. **Desain Sistem**
2. **Desain Sistem**

****

1. Tabel (JTable) untuk menampilkan data.
2. Total Payment (JLabel) Untuk menjumlahkan total sesuai data pada payment

**Kolom Input:**

1. Code (JTextField) → Auto-Generate

* Kode unik (Primary Key) otomatis untuk setiap memasukkan data. Tidak bisa diubah manual oleh pengguna.

1. Date (JDateChooser)

* Input tanggal menggunakan komponen tanggal. Mempermudah saat memilih tanggal dan memastikan format tanggal tetap kosisten saat pengisian atau diubah.

1. Page (JTextField)

* Input jumlah lembar yang dikerjakan

1. Author Name (JTextField)

* Input nama author yang lembarnya sedang dikerjakan

1. Job (JComboBox)

* Input jenis pekerjaan yang dilakukan. Dipilih dari daftar pilihan:
* **Color** → Jika hanya memberi warna basic
* **Shadow** → Jika hanya memberi efek shading/ bayangan
* **Color + Shadow** → Mengerjakan semua job warna basic dan bayangan

1. Payment (JTextField) → Format Rupiah

* Input jumlah pembayaran, diformat otomatis ke dalam format mata uang rupiah untuk memudahkan pembacaan.

**Tombol Aksi:**

1. ADD/ SAVE (JButton)

* **ADD** → Untuk menambahkan data baru
* **SAVE** → Untuk menyimpan data baru yang telah diisi ke database

1. EDIT/ SAVE (JButton)

* **EDIT** → Untuk mengedit data yang ada
* **SAVE** → Untuk menyimpan perubahan ke database

1. DELETE (JButton)

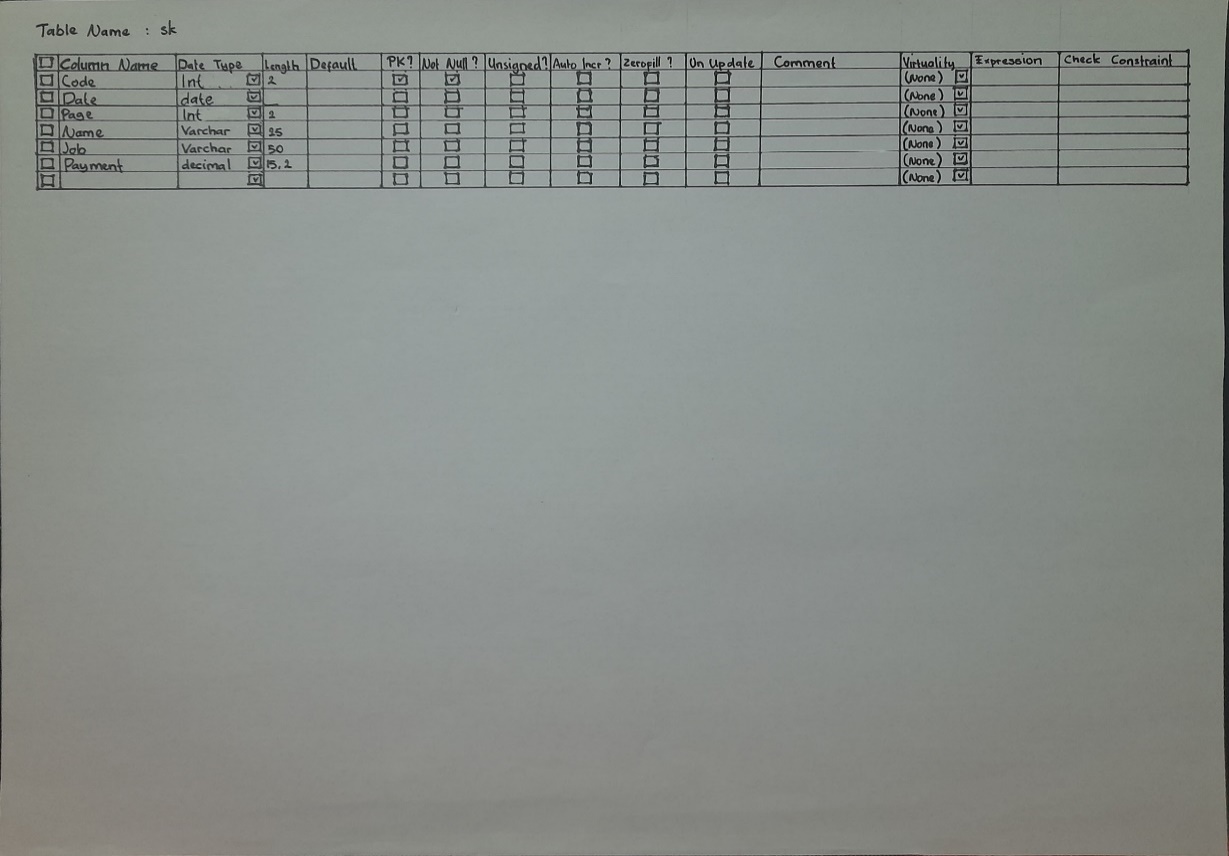
* Menghapus data yang dipilih dari database, dengan konfirmasi pengguna

1. CLOSE/ CANCEL (JButton)

* **CLOSE** → Menutup aplikasi, dengan konfirmasi pengguna
* **CANCEL** → Membatalkan proses ADD/ EDIT dan mengembalikan form ke posisi semula

**Cara Kerja Aplikasi:**

1. Tampilan Awal [JTable]
2. Saat aplikasi dijalankan, form akan terbuka dengan data langsung ditampilkan,
3. Kolom input data [JTextField] akan dalam kondisi tidak bisa diedit [false],
4. Tombol ADD [JButton] dalam kondisi hidup [true], dan untuk tombol EDIT & DELETE [JButton] dalam kondisi tidak bisa diklik [false]
5. Tampilan saat klik tombol ADD
6. Tekan tombol ADD [JButton] → isi form [JTextField] → tekan tombol SAVE [JButton] → data disimpan ke database → ditampilkan pada form [JTable] ditampilan awal
7. Untuk kolom input data [JTextField] akan dalam kondisi hidup [true]
8. Untuk tombol EDIT & DELETE [JButton] dalam kondisi tidak bisa diklik [false]
9. Untuk tombol CLOSE [JButton] berubah menjadi tombol CANCEL [JButton]
10. Tampilan saat klik tombol EDIT
11. Pilih data pada tabel → tekan tombol EDIT → ubah isi data → tekan tombol SAVE → data baru disimpan ke database → ditampilkan pada tabel
12. Kolom input data → true
13. Tombol EDIT → Tombol SAVE
14. Tombol ADD & DELETE → false
15. Tombol CLOSE → Tombol CANCEL
16. Tampilan saat klik tombol DELETE
17. Pilih data pada tabel → tekan tombol DELETE → muncul konfirmasi hapus → pilih yes → data pada database akan terhapus → tampilan tabel akan diperbaharui
18. Tampilan saat klik tombol CLOSE
19. Klik tombol CLOSE → muncul konfirmasi keluar aplikasi → selesai
20. **Desain Database**

****

Keterangan:

* Code → Type data ‘Int’ → Primay Key
* Date → Type data ‘Date’ → Menyimpan data tanggal dengan format (YYYY-MM-DD)
* Page → Type data ‘Varchar’ → Menyimpan data jumlah halaman
* Author Name → Type data ‘Varchar’ → Menyimpan data nama author [nama pembuat komik dari komik yang akan diedit karyawan]
* Job → Type data ‘Varchar’ → Menyimpan data dari kegiatan yang sudah dilakukan [Color, Shadow, dan Color + Shadow]
* Payment → Type data ‘Decimal’ → Menyimpan jumlah pembayaran dalam bentuk angka desimal agar mudah untuk dihitungatau dijumlahkan.