— Jollpi —
A Lightweight, Simple and Reliable Text Editor with Python 3, GTK4 and GtkSourceView 5

Zulfian

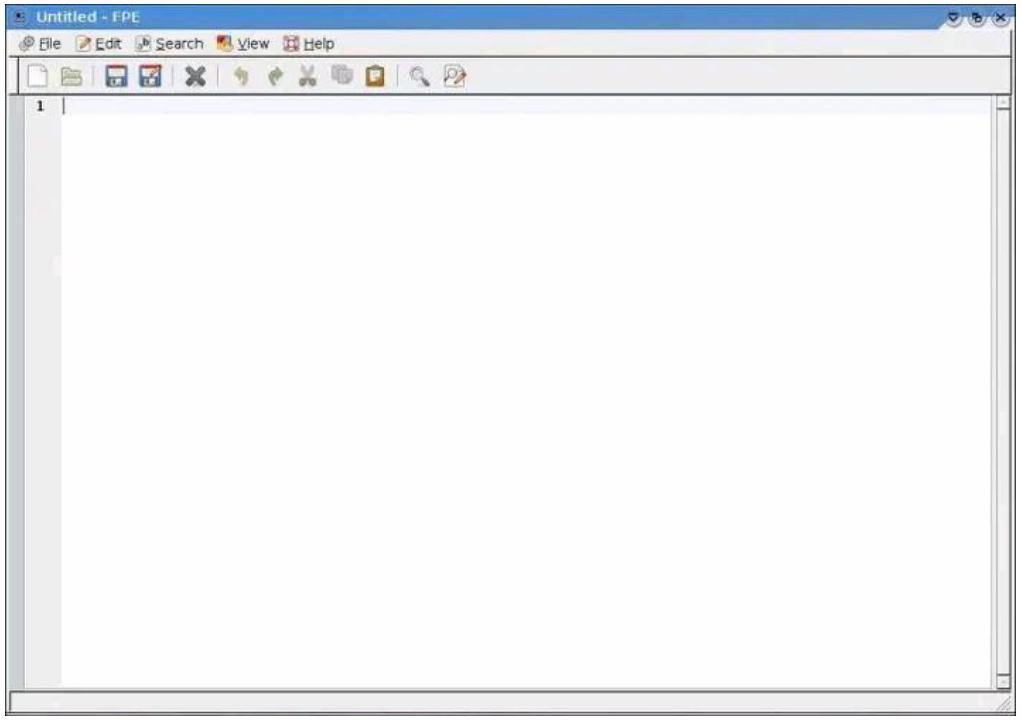
GNOME Asia Summit 2025 — Tokyo, Japan

14 December 2025

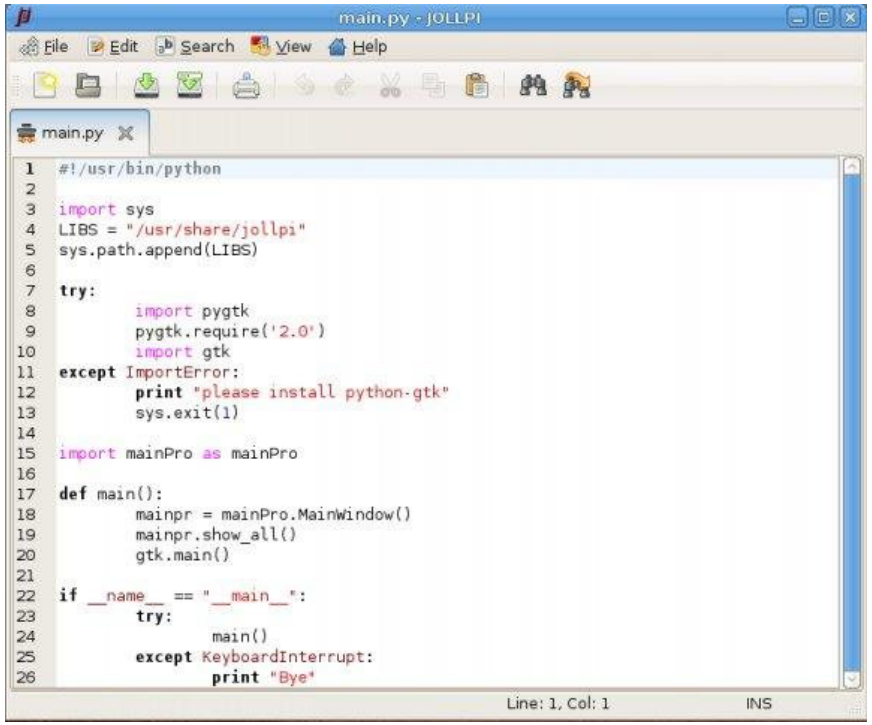
About Me

- Python Developer
- Linux user since 2008 (with a very long pause)
- Creator of Jollpi (2010 → rewrite 2025)

Flashback: Jollpi 2010



First prototype

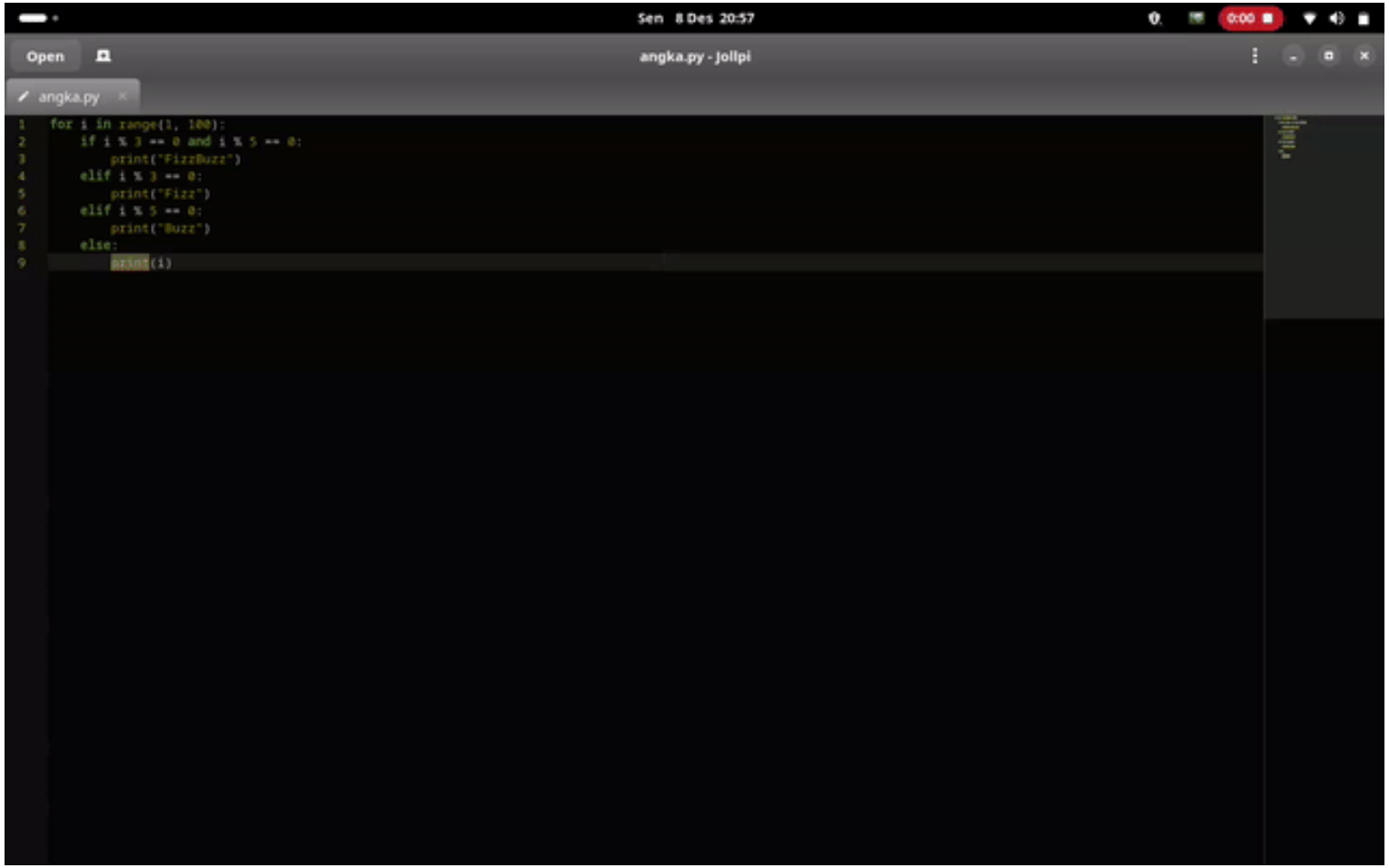


Last stable release

Python 2 — GTK2 — GtkSourceView 2 — Minimal — Multi-tab

» SourceForge: sourceforge.net/projects/jollpi
» GitLab: gitlab.com/zulfian1732/jollpi2-legacy

Jollpi 2025 — The Rewrite



Why Rewrite ?

- Python 2 is end-of-life
- GTK4 is a completely different world
- GtkSourceView 5 introduces a new approach
- User expectations changed
- Keep it light and minimal

Philosophy

- Simple and responsive — even with large files
- Only core components: Python, GTK4, GtkSourceView5
- Maintainable codebase
- Typing should feel effortless

Architecture

Python 3

GTK4 (GObject, GDK, GLib)

GtkSourceView 5

Gio File Handling + Monitoring

Async Loading

- ✔ Uses GtkSourceView.FileLoader
- ✔ Fully cancellable
- ✔ UI stays responsive while loading

```
loader = GtkSource.FileLoader.new(
    buffer,
    src_file
)
loader.load_async(
    priority,
    cancellable,
    None,
    None,
    on_loaded,
    ()
)
```

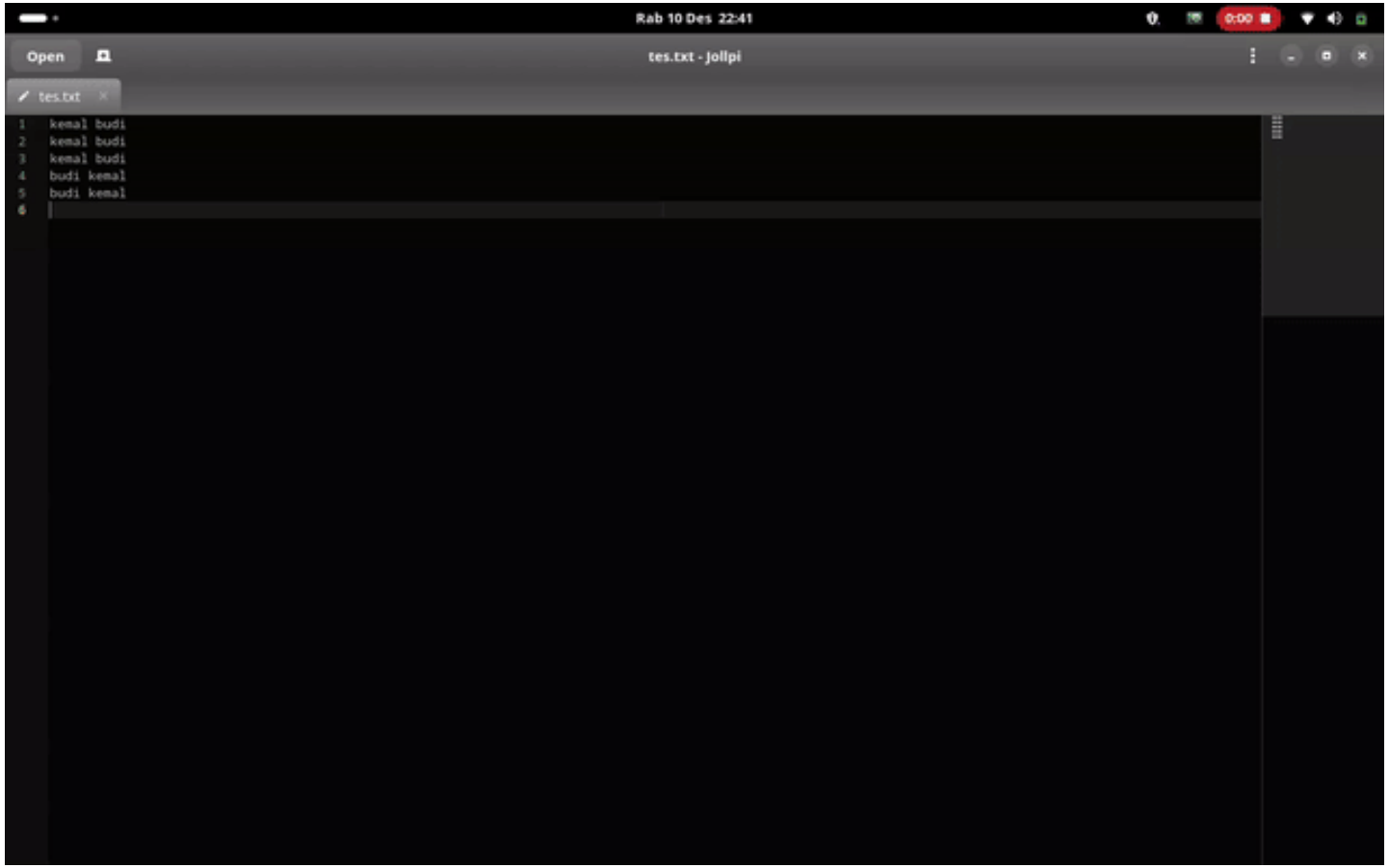
Async Search

- ✔ Async search using SearchContext
- ✔ Live result count
- ✔ Replace step-by-step without UI freezing

```
ctx = GtkSource.SearchContext.new(
    buffer,
    GtkSource.SearchSettings()
)
ctx.forward_async(
    iter,
    cancellable,
    on_match,
    None
)
```

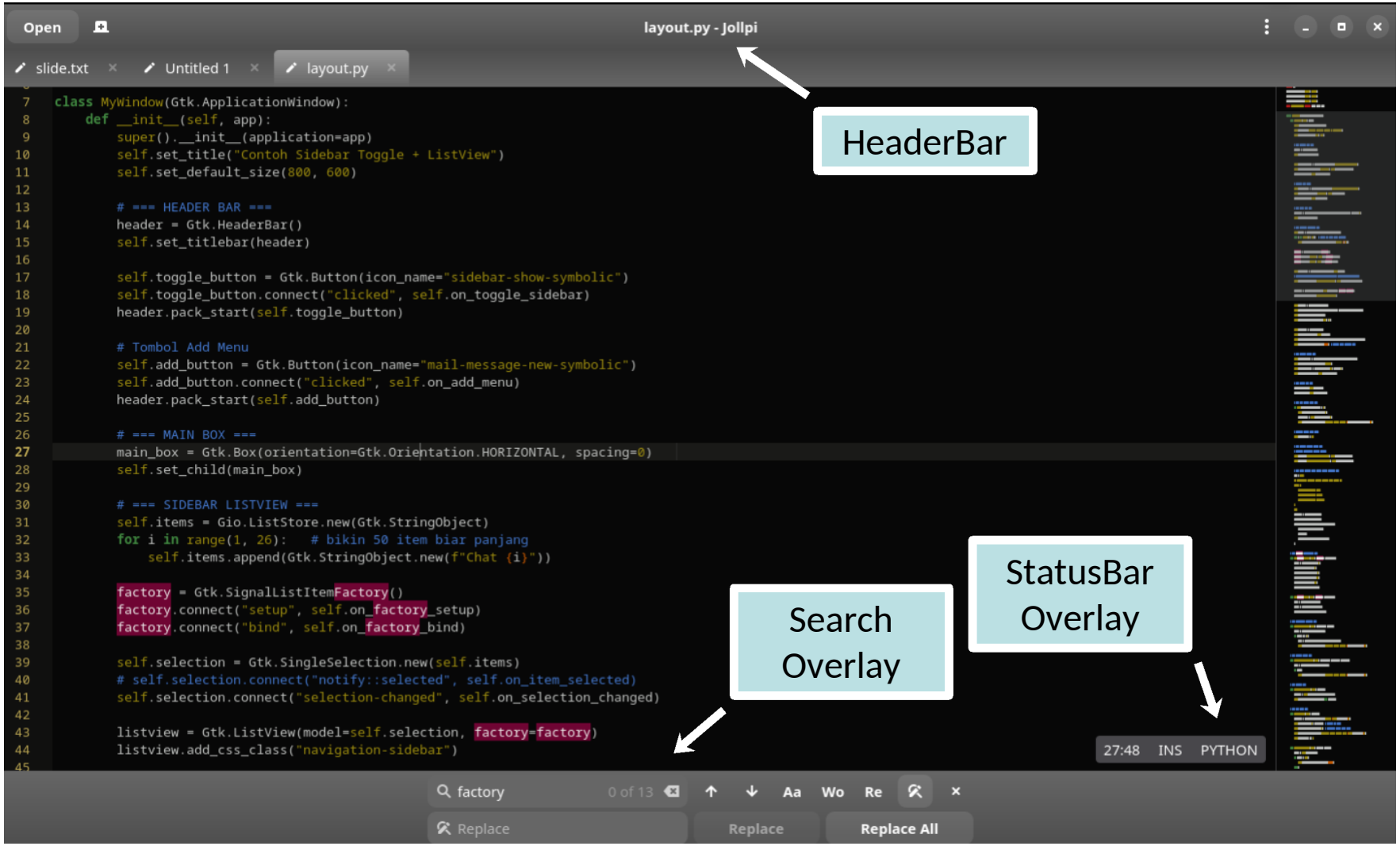
File Monitoring

- ✔ Uses Gio.FileMonitor
- ✔ Detects external modifications
- ✔ Offers reload / ignore options



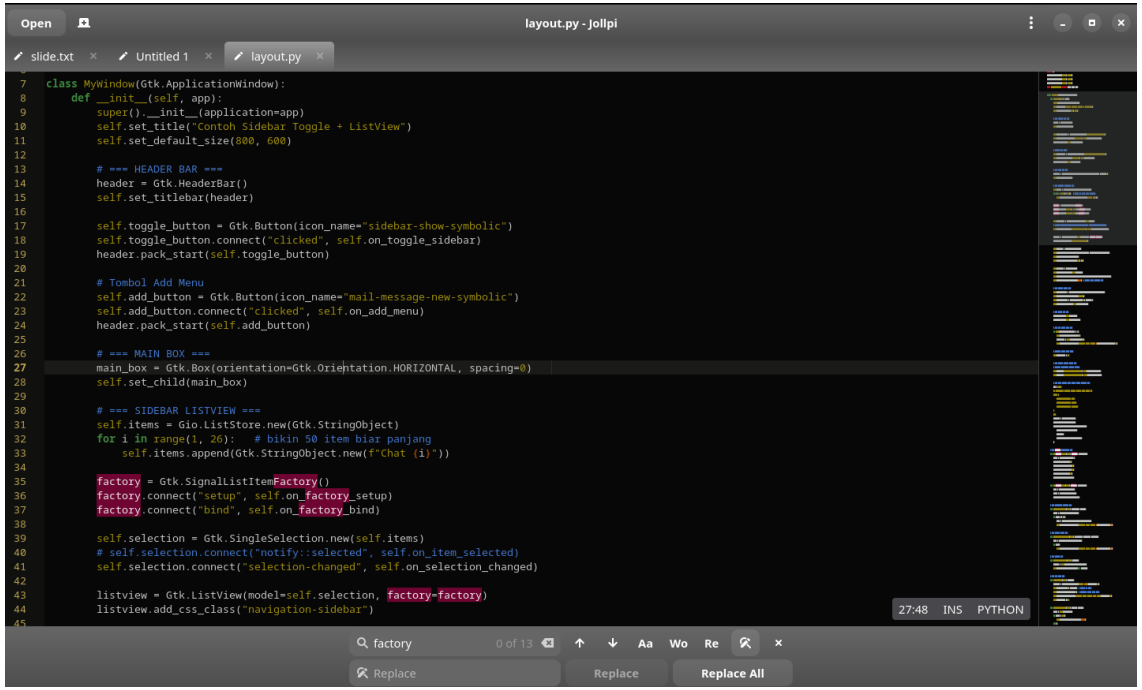

```
gfile = Gio.File.new_for_path(
    filepath
)
monitor = gfile.monitor_file(
    Gio.FileMonitorFlags.NONE,
    None
)  # returns Gio.FileMonitor
monitor.connect(
    "changed",
    on_file_changed
)
```

Modern GTK4 UI



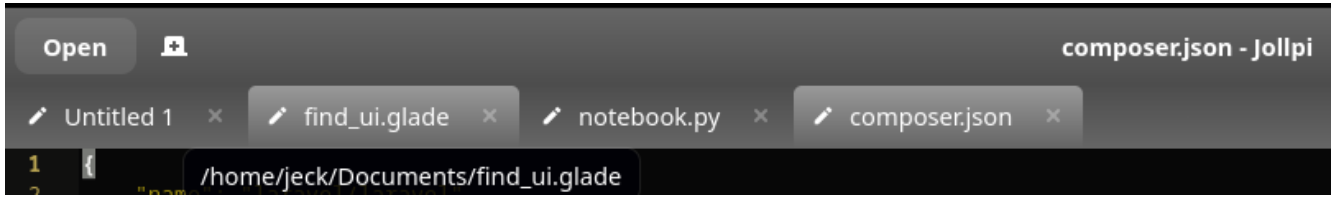
Custom Theme

```
<style-scheme id="jollpi" _name="Jollpi" version="1.0">

  <author>Zulfian</author>
  <_description>
  Just another scheme, which was taken and modified from oblivion
style
  </_description>

  <!--  Palette -->
  <color name="butter1"            value="#F9ED87"/>
  <color name="butter2"            value="#9F8E10"/>
  <color name="butter3"            value="#c4a000"/>
  <color name="chameleon1"         value="#8ae234"/>
  <!-- ......... -->
  <!-- Global Settings -->
  <style name="selection"          background="butter1"/>
  <style name="cursor"             foreground="aluminium2"/>
  <style name="secondary-cursor"   foreground="aluminium4-3-blend"/>
  <style name="current-line"       background="aluminium5"/>
  <style name="draw-spaces"        foreground="aluminium4"/>
  <style name="background-pattern" background="aluminium6-5-blend"/>
```

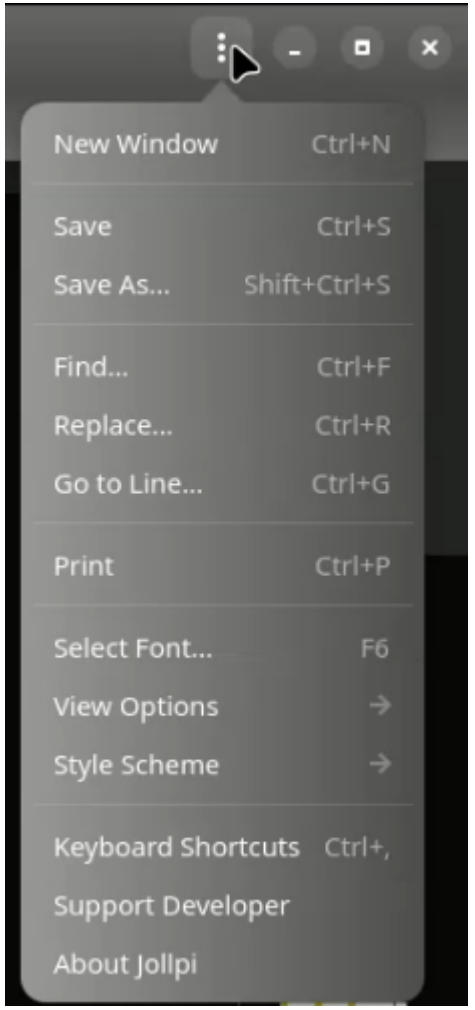


Optional custom theme inspired by Oblivion — modified and simplified.

GTK Custom CSS



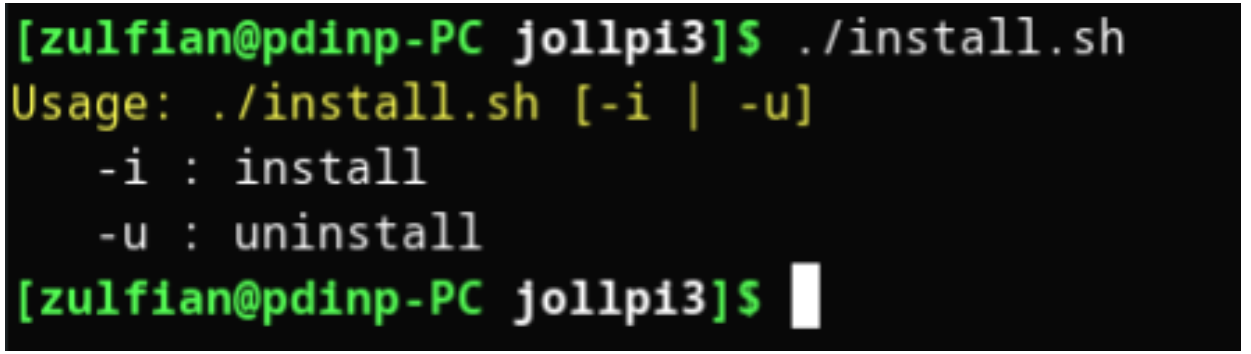
- Custom CSS via Gtk.CssProvider
- Consistent look and feel
- Lightweight — no extra libraries



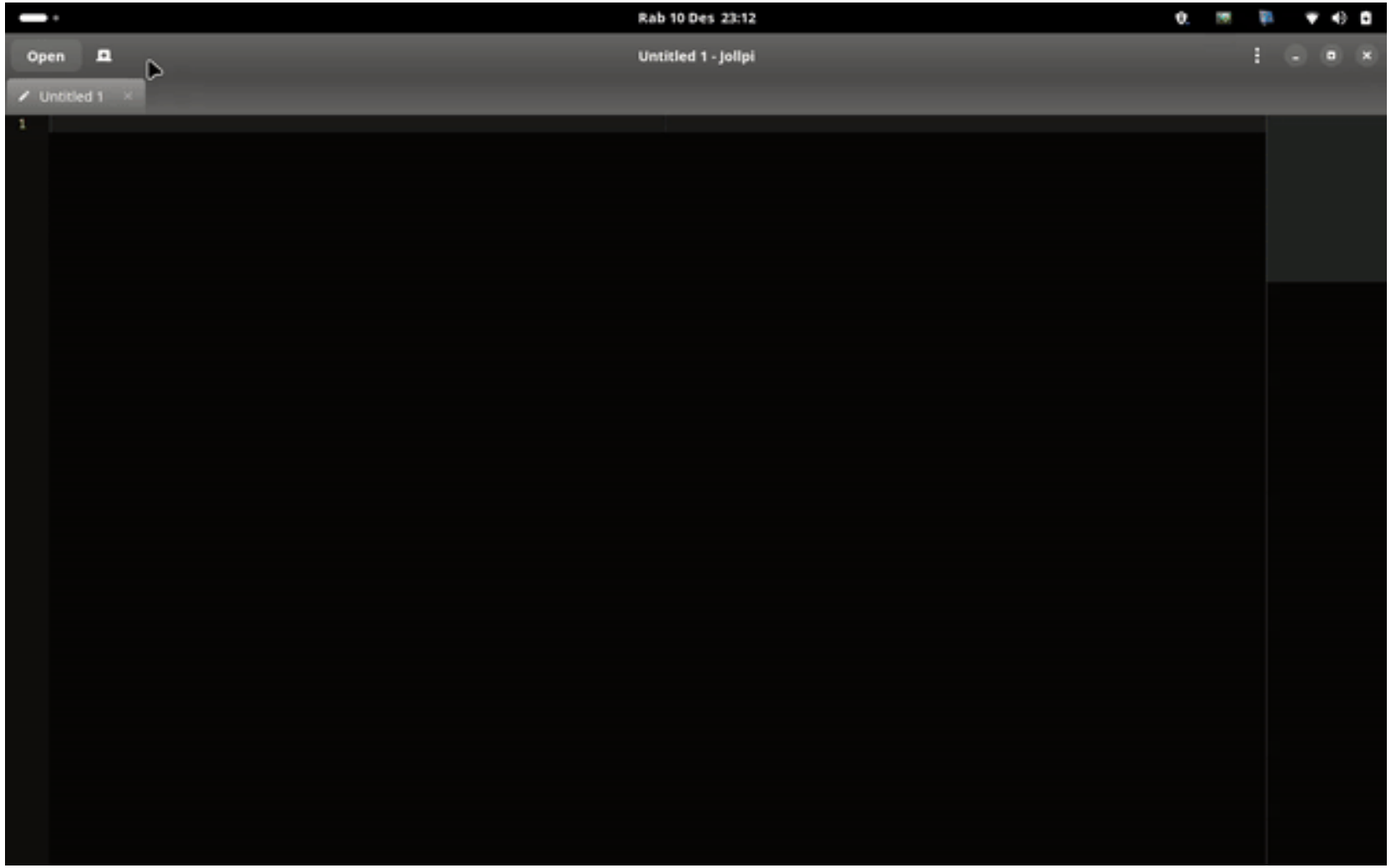

```
provider = gtk.CssProvider()
provider.load_from_file(custom.css)

gtk.StyleContext.add_provider_for_display(
    gdk.Display.get_default(),
    provider,
    gtk.STYLE_PROVIDER_PRIORITY_USER
)
```

Installer / install.sh

- ✔ Automated installation
- ✔ Simple, self-contained script
- ✔ Handles setup without extra tools

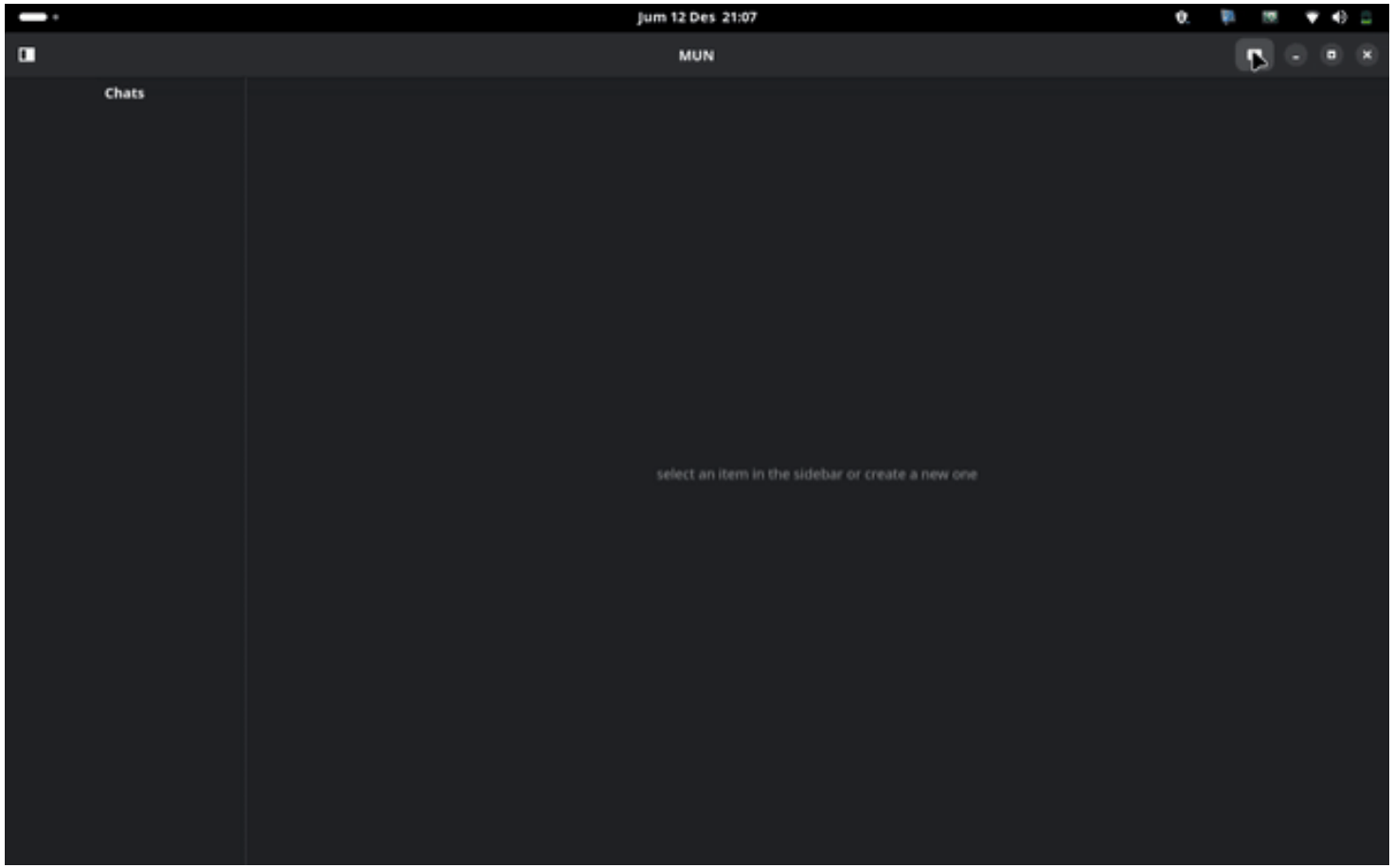


Demo — Jollpi (2025)



From Jollpi to MUN-AI

☑ Same toolkit ☑ Same lightweight philosophy ☑ Same maintainability mindset



Jollpi is now public

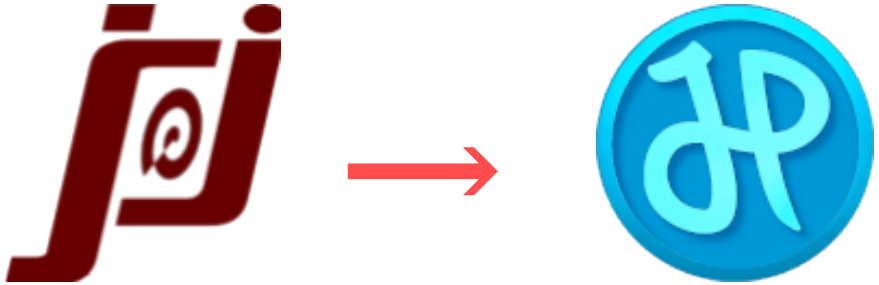
gitlab.com/zulfian1732/jollpi-text-editor

SCAN ME

Use it. Learn from it

~ Thank You ~

"Old code isn't failure. Sometimes it's just waiting for a better version of us."



» LinkedIn: ZULFIAN S.Kom

» GitLab: @zulfian1732

» GitHub: @zulfian1732

» Mastodon: @zulfian

» E-mail: zulfian1732@gmail.com