

LAPORAN PRAKTIKUM
ALGORITMA PEMOGRAMAN
PEMROGRAMAN GUI 2 (PEMBUATAN KALKULATOR) PADA
BAHASA PEMROGRAMAN JAVA

Disusun Oleh :

Aliifah Felda Mufarrihati Salwaa

2511531011

Dosen Pengampu :

Dr. Wahyudi, S.T., M.T.

Asisten Praktikum :

Aufan Taufiqurrahman



DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
TAHUN 2025

KATA PENGANTAR

Laporan praktikum ini disusun sebagai salah satu bentuk pertanggungjawaban kegiatan praktikum algoritma pemrograman yang membahas tentang pemrograman GUI pada bahasa Java. Melalui laporan ini penulis dapat memahami materi praktikum secara mendalam dan dapat lebih teliti, teratur, serta memiliki kemampuan yang baik dalam penulisan kode pemrograman sesuai kaidah akademik. Sehingga laporan ini dapat menjadi sarana belajar, dokumentasi kegiatan, dan referensi praktikum berikutnya.

Penulis menyadari bahwa laporan ini masih memiliki banyak kekurangan, baik dari isi maupun penyajiannya. Oleh karena itu, saran dan kritik sangat penulis harapkan guna untuk menyempurnakan laporan berikutnya.

Padang, 26 November 2025

Aliifah Felda Mufarrihati Salwaa

2511531011

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Praktikum	1
1.2 Tujuan Praktikum.....	1
1.3 Manfaat Praktikum.....	2
BAB II PEMBAHASAN	3
2.1 Praktikum Desain GUI Kalkulator dengan WindowBuilder.....	3
2.2 Praktikum Pemrograman GUI Kalkulator dengan WindowBuilder ...	4
2.3 Output Pemrograman Kalkulator dengan WindowBuilder	11
BAB III KESIMPULAN	13
3.1 Kesimpulan.....	13
3.2 Saran.....	13
DAFTAR PUSTAKA	14

BAB I

PENDAHULUAN

1.1 Latar Belakang Praktikum

Seperti yang kita ketahui, saat ini GUI (*Graphical User Interface*) merupakan hal yang sangat penting dalam pemrograman untuk pengembangan aplikasi ataupun website. GUI sangat dibutuhkan untuk mempermudah interaksi manusia dengan perangkat. Salah satu contohnya adalah dalam operasi aritmatika. Terkadang penjumlahan dengan angka yang besar membutuhkan waktu yang lama jika diselesaikan secara manual. Oleh karena itu, salah satu aplikasi dari pemrograman GUI adalah kalkulator.

Dengan mempelajari pemrograman GUI, kita dapat membuat sebuah kalkulator menggunakan desain *interface* yang sudah dikenali banyak orang. Hal ini membuat kalkulator kita dapat dimengerti dan digunakan oleh orang awam sekalipun. Disini mahasiswa diajarkan bagaimana cara untuk membuat *inteface* yang baik dan kode program yang dapat menjalankan *interface* tersebut sehingga dapat digunakan dengan lancar. Pemrograman GUI dengan membuat kalkulator ini tentunya menjadi sebuah jembatan bagi mahasiswa untuk lebih memahami bagaimana sistem program GUI dengan menggunakan Java Swing dan WindowBuilder pada Eclipse.

1.2 Tujuan Praktikum

Tujuan dari pelaksanaan praktikum ini adalah sebagai berikut:

1. Mengetahui cara membuat program GUI kalkulator dengan menggunakan Java Swing pada WindowBuilder Eclipse.
2. Mengetahui langkah-langkah untuk membuat GUI kalkulator.
3. Mengetahui hasil dari pemrograman GUI kalkulator dengan bahasa pemrograman Java.

1.3 Manfaat Praktikum

Manfaat dari pelaksanaan praktikum ini adalah sebagai berikut:

1. Memahami bagaimana cara pengembangan GUI dan pengaplikasian pemrograman GUI.
2. Mampu mengimplementasikan pemrograman GUI.
3. Meningkatkan kemampuan dalam penggunaan bahasa pemrograman Java.

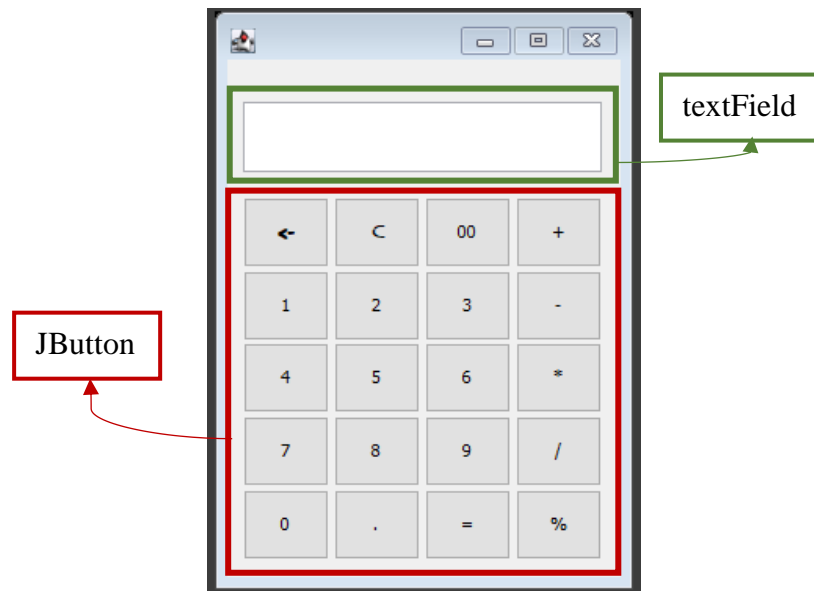
BAB II PEMBAHASAN

2.1. Praktikum Desain GUI Kalkulator dengan WindowBuilder

Pada praktikum kali ini kita akan kembali menggunakan Eclipse WindowBuilder untuk membuat *interface* yang kita inginkan. Seperti yang telah kita ketahui dari praktikum sebelumnya, bahwasanya penggunaan Eclipse WindowBuilder ini dapat mempermudah kita dalam membuat desain *interface* tanpa harus melakukan koding secara manual. Pada praktikum kali ini kita akan menggunakan Eclipse WindowBuilder untuk membuat program GUI kalkulator.

Berikut adalah penjelasan lebih lanjut tentang pembuatan desain GUI kalkulator menggunakan Eclipse WindowBuilder:

1. Pertama-tama kita akan membuat class dengan WindowBuilder yang menggunakan Java Swing.
2. Selanjutnya kita akan melakukan pembuatan desain dari kalkulator dengan menggunakan bagian desain pada tampilan eclipse.



Gambar 2.1: Desain Kalkulator

3. Pertama kita akan memasukkan banyak JButton ke dalam desain GUI kita. Seperti yang kita ketahui dari praktikum sebelumnya, JButton berfungsi untuk menjalankan sebuah program ketika pengguna melakukan klik pada tombol tersebut. Disini kita menggunakan banyak JButton selayaknya banyak tombol yang ada pada kalkulator, namun kita hanya memasukkan operasi penjumlahan, pengurangan, perkalian, pembagian, dan modulus.
4. Kemudian kita akan memasukkan textField yang berfungsi untuk menampilkan input dari pengguna. Selayaknya kalkulator, textField ini nantinya akan menampilkan angka input dan output.

Pada bagian desain, pembuatan kalkulator termasuk mudah untuk diciptakan. Untuk memasukkan JButton yang berulang dapat kita lakukan hanya dengan *copy-paste* dari JButton pertama. Kemudian kita hanya perlu untuk mengganti text dari masing-masing tombol dan mengubah nama variabel tombol dengan nama yang sesuai. Selain itu, kita juga hanya membutuhkan satu textField untuk menampilkan input dan output ketika kalkulator digunakan.

2.2. Praktikum Pemrograman GUI Kalkulator dengan WindowBuilder

Setelah melakukan desain, tentunya kita membutuhkan kode program agar kalkulator yang telah kita desain dapat berfungsi dengan baik dan benar. Karena kita menambahkan banyak button, kita akan mendapatkan banyak kode program yang otomatis muncul pada bagian *source code*. Kode-kode ini yang nantinya akan kita modifikasi dan akan kita tambah agar kalkulator kita dapat berfungsi.

Berikut adalah langkah-langkah dalam membuat kode program GUI kalkulator:

1. Pertama kita akan melakukan modifikasi program untuk bagian angka agar angka yang di klik oleh pengguna dapat tampil pada kalkulator. Kita akan melakukan modifikasi pada angka '00' terlebih dahulu.

```

JButton btnNumber00 = new JButton("00");
btnNumber00.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

```

```

textField.setText(textField.getText()+btnNumber00.getText());
    }
});
btnNumber00.setBounds(126, 88, 55, 45);
contentPane.add(btnNumber00);

```

Kode Program 2.1: Pemrograman angka “00”

Setelah kita melakukan *double-click* pada tombol ‘00’ kita akan mendapatkan kode action seperti yang tertera di bawah.

```

btnNumber00.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
    }
}

```

Agar tombol dapat berfungsi kita akan memasukkan rangkaian kode ke dalam method `actionPerformed(ActionEvent e)`. Pada kasus angka ini, kita akan memasukkan kode program seperti yang tertera pada kode program 2.1. Kode program tersebut berfungsi untuk memanggil text “00” dan menampilkannya pada bagian `textField` yang tersedia. Pemrograman seperti ini dapat kita lakukan berulang untuk angka-angka lainnya. Berikut adalah pemrograman lengkap untuk bagian angka:

```

JButton btnNumber00 = new JButton("00");
btnNumber00.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

textField.setText(textField.getText()+btnNumber00.getText());
    }
});
btnNumber00.setBounds(126, 88, 55, 45);
contentPane.add(btnNumber00);

JButton btnNumber0 = new JButton("0");
btnNumber0.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

textField.setText(textField.getText()+btnNumber0.getText());
    }
});
btnNumber0.setBounds(10, 275, 55, 45);
contentPane.add(btnNumber0);

JButton btnNumber1 = new JButton("1");
btnNumber1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

textField.setText(textField.getText()+btnNumber1.getText());
    }
}

```



```

    });
    btnNumber1.setBounds(10, 135, 55, 45);
    contentPane.add(btnNumber1);

    JButton btnNumber2 = new JButton("2");
    btnNumber2.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {

textField.setText(textField.getText()+btnNumber2.getText());
        }
    });
    btnNumber2.setBounds(68, 135, 55, 45);
    contentPane.add(btnNumber2);

    JButton btnNumber3 = new JButton("3");
    btnNumber3.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {

textField.setText(textField.getText()+btnNumber3.getText());
        }
    });
    btnNumber3.setBounds(126, 135, 55, 45);
    contentPane.add(btnNumber3);

    JButton btnNumber4 = new JButton("4");
    btnNumber4.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {

textField.setText(textField.getText()+btnNumber4.getText());
        }
    });
    btnNumber4.setBounds(10, 181, 55, 45);
    contentPane.add(btnNumber4);

    JButton btnNumber5 = new JButton("5");
    btnNumber5.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {

textField.setText(textField.getText()+btnNumber5.getText());
        }
    });
    btnNumber5.setBounds(68, 181, 55, 45);
    contentPane.add(btnNumber5);

    JButton btnNumber6 = new JButton("6");
    btnNumber6.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {

textField.setText(textField.getText()+btnNumber6.getText());
        }
    });
    btnNumber6.setBounds(126, 181, 55, 45);
    contentPane.add(btnNumber6);

    JButton btnNumber7 = new JButton("7");
    btnNumber7.addActionListener(new ActionListener() {

```

```

        public void actionPerformed(ActionEvent e) {
            textField.setText(textField.getText()+btnNumber7.getText());
        }
    });
    btnNumber7.setBounds(10, 228, 55, 45);
    contentPane.add(btnNumber7);

    JButton btnNumber8 = new JButton("8");
    btnNumber8.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {

            textField.setText(textField.getText()+btnNumber8.getText());
        }
    });
    btnNumber8.setBounds(68, 228, 55, 45);
    contentPane.add(btnNumber8);

    JButton btnNumber9 = new JButton("9");
    btnNumber9.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {

            textField.setText(textField.getText()+btnNumber9.getText());
        }
    });
    btnNumber9.setBounds(126, 228, 55, 45);
    contentPane.add(btnNumber9);

```

Kode Program 2.2: Pemrograman angka

2. Selanjutnya, kita akan membuat logika program untuk operasi aritmatika pada kalkulator. Sama seperti sebelumnya, kita hanya perlu melakukan *double-click* agar dapat diarahkan langsung pada kode tombol yang ingin kita modifikasi. Untuk contoh operasi pertama, disini kita akan membuat kode program untuk proses perkalian. Berikut adalah kode program yang akan kita masukkan:

```

JButton btnKali_2511531011 = new JButton("*");
btnKali_2511531011.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        num1 = Double.parseDouble(textField.getText());
        operator = '*';
        textField.setText("");
    }
});
btnKali_2511531011.setBounds(184, 181, 55, 45);
contentPane.add(btnKali_2511531011);

```

Kode Program 2.3: Pemrograman operator perkalian

Disini ketika pengguna menekan tombol operasi “*”, angka pertama yang dimasukkan pengguna (yang telah tampil pada textField) akan diambil dan disimpan di dalam variabel num1. Sementara itu, variabel operator akan kita set menjadi operasi penjumlahan dan textField akan dikosongkan. Sama seperti kode program angka sebelumnya, kita juga bisa melakukan *copy-paste* untuk kode program operator ini dan hanya mengubah sesuai dengan operator apa yang tertera pada setiap tombol. Berikut adalah seluruh kode program bagian kode operator:

```

        JButton btnPlus_2511531011 = new JButton("+");
        btnPlus_2511531011.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                num1 =
Double.parseDouble(textField.getText());
                operator = '+';
                textField.setText("");
            }
        });
        btnPlus_2511531011.setBounds(184, 88, 55, 45);
        contentPane.add(btnPlus_2511531011);

        JButton btnMinus_2511531011 = new JButton("-");
        btnMinus_2511531011.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                num1 =
Double.parseDouble(textField.getText());
                operator = '-';
                textField.setText("");
            }
        });
        btnMinus_2511531011.setBounds(184, 135, 55, 45);
        contentPane.add(btnMinus_2511531011);

        JButton btnKali_2511531011 = new JButton("*");
        btnKali_2511531011.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                num1 =
Double.parseDouble(textField.getText());
                operator = '*';
                textField.setText("");
            }
        });
        btnKali_2511531011.setBounds(184, 181, 55, 45);
        contentPane.add(btnKali_2511531011);

        JButton btnBagi_2511531011 = new JButton("/");
        btnBagi_2511531011.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                num1 =
Double.parseDouble(textField.getText());

```

```

        operator = '/';
        textField.setText("");
    }
});
btnBagi_2511531011.setBounds(184, 228, 55, 45);
contentPane.add(btnBagi_2511531011);

JButton btnMod_2511531011 = new JButton("%");
btnMod_2511531011.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        num1 =
Double.parseDouble(textField.getText());
        operator = '%';
        textField.setText("");
    }
});
btnMod_2511531011.setBounds(184, 275, 55, 45);
contentPane.add(btnMod_2511531011);

```

Kode Program 2.4: Pemrograman operator

3. Selanjutnya kita akan melakukan set kode program untuk tombol tambahan bagian *delete*. Untuk tombol *delete* ini, kita menggunakan perulangan *for* untuk menghapus satu angka paling akhir yang ditampilkan oleh textField dengan mengurangi panjang dari isi textField. Berikut adalah kode program dari tombol *delete*:

```

JButton btndel = new JButton("<-");
btndel.setFont(new Font("Snap ITC", Font.BOLD, 11));
btndel.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String string = textField.getText();
        textField.setText("");
        for (int i=0; i<string.length()-1;i++) {

textField.setText(textField.getText()+string.charAt(i));
        }
    }
});
btndel.setBounds(10, 88, 55, 45);
contentPane.add(btndel);

```

Kode Program 2.5: Pemrograman *delete*

4. Selanjutnya kita akan melakukan set program untuk tombol tambahan lainnya yakni *clear*. Tombol *clear* ini bertujuan untuk menghapus seluruh isi dari textField. Berikut adalah kode program dari tombol *clear*:

```

JButton btnclear = new JButton("C");
btnclear.setFont(new Font("Tempus Sans ITC", Font.BOLD, 11));

```

```

btnclear.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        textField.setText("");
    }
});
btnclear.setBounds(68, 88, 55, 45);
contentPane.add(btnclear);

```

Kode Program 2.6: Pemrograman *clear*

5. Tombol tambahan terakhir yang akan kita masukkan adalah tombol koma. Tombol koma ini bertujuan untuk membuat bilangan desimal sehingga nantinya kita juga dapat mengoperasikan bilangan desimal. Berikut adalah kode program dari tombol koma:

```

JButton btnKoma = new JButton(".");
btnKoma.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        textField.setText(textField.getText().concat("."));
    }
});

btnKoma.setBounds(68, 275, 55, 45);
contentPane.add(btnKoma);

```

Kode Program 2.7: Pemrograman koma

6. Langkah terakhir dan paling penting adalah untuk memasukkan logika aritmatika agar operasi dapat dilakukan. Logika ini akan kita masukkan ke dalam tombol “=” menggunakan *switch-case* sehingga ketika pengguna menekan tombol ini, operasi aritmatika akan terjadi sesuai dengan operator yang dipilih. Berikut adalah kode program untuk tombol “=”:

```

JButton btnSD = new JButton("=");
btnSD.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        num2 = Double.parseDouble(textField.getText());

        switch (operator) {
            case '+':
                result=num1+num2;
                break;

            case '-':
                result=num1-num2;
                break;

            case '*':
                result=num1*num2;
                break;

```

```

        case '/':
            result=num1/num2;
            break;

        case '%':
            result=num1%num2;
            break;
    }
    textField.setText(String.valueOf(result));
    num1=result;
}
});
btnSD.setBounds(126, 275, 55, 45);
contentPane.add(btnSD);

```

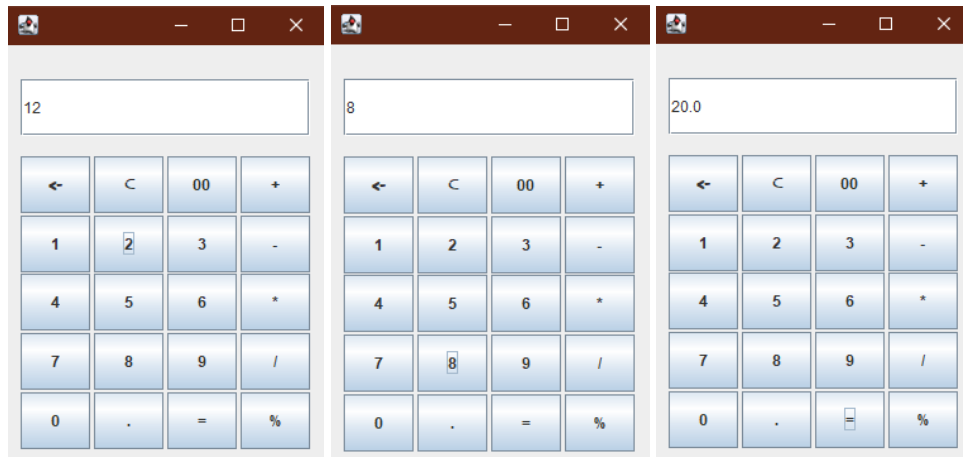
Kode Program 2.8: Pemrograman logika aritmatika

Pada awal kita akan mengambil isi kedua dari textField (angka yang di input setelah menekan tombol operator) dan menyimpannya pada variabel num2. Otomatis pada tahap ini kita sudah memiliki 2 variabel yang dapat kita operasikan. Kemudian program akan masuk pada bagian logika aritmatika dengan menggunakan *switch-case* dimana masing masing operasi akan berbeda sesuai dengan operator yang telah dipilih. Terakhir kita akan menampilkan hasil operasi pada textField dan kembali set hasil tersebut ke variabel num1 agar operasi dalam kalkulator dapat dilakukan secara berkelanjutan.

Dengan semua modifikasi yang sudah kita lakukan pada *source code* ini, seharusnya program kalkulator kita sekarang dapat berjalan dengan lancar. Pengguna sudah bisa melakukan berbagai operasi aritmatika, menghapus dan memperbaiki input, hingga melakukan operasi secara berkelanjutan tanpa harus melakukan input berulang.

2.3. Output Pemrograman Kalkulator dengan WindowBuilder

Selanjutnya kita akan melakukan tes pada program kita. Sebagai contoh kita akan memasukkan input pertama angka 12, operator penjumlahan, dan input kedua angka 8.



Gambar 2.2: Output pemrograman kalkulator

Dari sini kita dapat melihat bahwasanya program kita sudah berjalan dengan baik untuk operator penjumlahan. Kita juga bisa memastikan satu-persatu tombol operator apakah sudah berjalan dengan baik atau belum. Kita juga dapat melakukan uji coba untuk semua tombol tambahan seperti *delete*, *clear*, dan melakukan operasi dengan bilangan desimal.

BAB III

KESIMPULAN

3.1. Kesimpulan

Pemrograman kalkulator pada praktikum kali ini merupakan pemrograman dasar yang mengimplementasikan materi GUI (*Graphical User Interface*). Pemrograman kalkulator ini merupakan langkah untuk semakin memahami GUI, logika pemrograman, dan algoritma pemrograman. Disini mahasiswa ditantang untuk membuat kode program sedemikian rupa agar kalkulator dapat berjalan dengan baik. Dari sini mahasiswa dapat semakin memahami bagaimana cara kerja pemrograman GUI dan bagaimana cara untuk memodifikasi *source code* agar program berjalan sesuai dengan yang kita inginkan.

3.2. Saran

Praktikum kali ini merupakan latihan baru agar kita lebih mahir dalam pemrograman GUI dan logika pemrograman. Kalkulator yang kita buat sekarang dapat disebut sebagai proyek kecil yang bisa membantu kita dalam belajar pemrograman GUI kedepannya. Proyek ini juga dapat kita gunakan sebagai evaluasi untuk pemahaman kita terhadap kode program dan pemrograman GUI. Dari sini kita dapat belajar dan mengetahui dimana kelemahan kita dan diharapkan kita dapat memperbaikinya untuk proyek-proyek mendatang.

DAFTAR PUSTAKA

- [1] H. Schildt, *Java: A Beginner's Guide*, 6th ed. New York, NY: McGraw-Hill Education, 2014
- [2] J. David, *Introduction to Programming Using Java*, 9th ed. New York, NY, 2022
- [3] "GUI Calculator in JAVA using eclipse ide," YouTube, dipublikasikan oleh Tech Projects, Apr. 03, 2020. [Daring]. Tersedia pada: <https://www.youtube.com/watch?v=mZVD5OssHQM>. [Diakses: Nov. 24, 2025].