

Rumpun Ilmu: Teknik Sipil

LAPORAN AKHIR PENELITIAN



PODOMORO
UNIVERSITY

**Penggunaan Bahasa Pemrograman Python untuk Membuat
Aplikasi yang Dapat Digunakan Sebagai Media Dalam
Mengajarkan Konsep Perilaku Lentur Balok Beton Bertulang**

TIM PENELITI
Suwito (0310036702)

Dibiayai Oleh
Mandiri

UNIVERSITAS AGUNG PODOMORO
JULI 2021

**PENGESAHAN LAPORAN AKHIR PENELITIAN
TAHUN 2021**

1. Judul Penelitian : Penggunaan Bahasa Pemrograman Python untuk Membuat Aplikasi yang Dapat Digunakan Sebagai Media Dalam Mengajarkan Konsep Perilaku Lentur Balok Beton Bertulang
2. Rumpun Ilmu : Teknik Sipil
3. Ketua Peneliti
- a. Nama Lengkap (Gelar) : Suwito, S.T., Ph.D.
- b. NIDN/NIP : 0310036702/2316017
- c. JJA/Pangkat : Lektor/IIIC
- d. Program Studi : Manajemen & Rekayasa Konstruksi
4. Anggota Peneliti :

No	Nama	Bidang Keahlian	Program Studi
1	-	-	-

5. Realisasi Durasi Penelitian : 3 (tiga) bulan
6. Realisasi Biaya : -

Jakarta, 9 Juli 2021

Mengetahui
Ketua Program Studi MRK



(Ferdinand Fassa, ST., MT.)
0401127905

Ketua Peneliti



(Suwito, ST., Ph.D.)
0310036702

Menyetujui,
Kepala LPPM



(Prof. Dr. Ir. Sony Heru Priyanto, MM.)
0614096601

Penggunaan Bahasa Pemrograman Python untuk Membuat Aplikasi yang Dapat Digunakan Sebagai Media Dalam Mengajarkan Konsep Perilaku Lentur Balok Beton Bertulang

ABSTRAK

Studi ini menyajikan pengembangan sebuah aplikasi komputer untuk menganalisis dan menggambarkan grafik dari sebuah potongan penampang balok beton bertulang yang mengalami pembebanan lentur atau momen. Bahasa pemrograman yang digunakan adalah Python, yang merupakan sebuah bahasa pemrograman yang populer, mudah dipelajari dan didukung berbagai modul untuk berbagai bidang termasuk bidang keteknikan. Dalam studi ini diperlihatkan penggunaan modul Matplotlib untuk membantu pembuatan kurva/grafik dan modul Tkinter untuk memudahkan pembuatan *Graphical User Interface* (GUI). Selain itu, sebagai bahasa pemrograman *interpreted*, Python memudahkan pengembangan aplikasi aplikasi bisa langsung dijalankan tanpa perlu *compiling* terlebih dahulu. Aplikasi yang dikembangkan adalah pembuatan grafik kurvatur versus momen. Perhitungan kurvatur dan momen dilakukan pada titik-titik yang mewakili perubahan mekanisme lentur dari potongan penampang balok. Aplikasi berhasil dikembangkan dan berjalan dengan baik. Aplikasi ini dapat digunakan oleh mahasiswa/i untuk mepedalam pengertiannya mengenai perilaku balok beton bertulang yang mengalami beban lentur.

Kata kunci: *Python, lentur, momen, balok beton bertulang.*

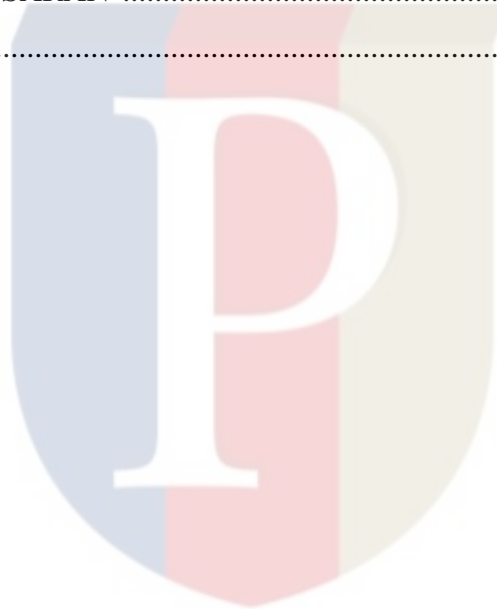
ABSTRACT

This studi presents a computer application development that can be used to analyze and draw graph of cross-section of reinforced concrete beam loaded with flexural moment. The programming language used in this study is Python, a popular, easy programming language that is supported with many libraries. In this study, it is demonstrated the use of Matplotlib library for plotting curve and Tkinter library to support *Graphical User Interface* (GUI) development. In addition, as interpreted language, application can be developed faster as the application can be run without compiling first. The developed application is for making graph of curvature versus moment. Curvature and moment are calculated at points representing the change of flexural mechanism on reinforced concrete beam cross-section. The application is successfully developed dan runs properly. This application can be used by students for improving their understanding on behavior of reinforced concrete beam experiencing flexural load.

Keywords: *Python, flexural, moment, reinforced concrete beam.*

DAFTAR ISI

HALAMAN SAMPUL	i
HALAMAN PENGESAHAN.....	ii
ABSTRAK	iii
DAFTAR ISI.....	iv
BAB I PENDAHULUAN	1
BAB II TINJAUAN PUSTAKA.....	3
BAB III METODE PENELITIAN	7
BAB IV HASIL DAN PEMBAHASAN	10
BAB V SIMPULAN DAN SARAN	11
DAFTAR PUSTAKA	12
LAMPIRAN	
Lampiran 1	13
Lampiran 2	13
Lampiran 3	13
Lampiran 4	16
Lampiran 5	16



BAB I

PENDAHULUAN

1.1 Latar Belakang

Material konstruksi yang paling banyak digunakan di Indonesia adalah beton, yang umumnya diaplikasikan dalam bentuk beton bertulang. Penggunaan beton bertulang dapat ditemukan pada segala bentuk komponen struktur seperti pelat, balok, kolom, dinding dan fondasi. Jadi dapat dikatakan bahwa material beton bertulang mempunyai peran yang sangat penting di industri konstruksi Indonesia. Oleh karena itu, pengertian yang kuat dalam teori dasar desain beton bertulang adalah sangat penting.

Mata kuliah desain beton bertulang merupakan subjek utama yang diajarkan pada program studi yang berhubungan dengan teknik sipil. Materi yang diberikan dalam kelas desain beton bertulang biasanya meliputi bagaimana cara menganalisis dan mendesain komponen struktur terhadap momen, gaya aksial, gaya geser dan torsi. Umumnya perilaku balok beton bertulang terhadap lentur merupakan hal mendasar yang wajib dipahami oleh para mahasiswa/i. Mekanisme yang terjadi pada potongan penampang balok beton bertulang dari awal pembebanan lentur sampai batas ultimit perlu dimengerti dengan baik. Namun perhitungan ini memerlukan waktu yang lama dan perlu dilakukan ulang ketika kita menganalisis kasus lain. Oleh karena itu, penggunaan bahasa pemrograman untuk membuat aplikasi yang dapat membantu kita lebih memahami suatu theory yang ada sering dilakukan. Keuntungan dari mengembangkan aplikasi secara mandiri adalah aplikasi dapat dibuat sesuai dengan keinginan.

1.2 Perumusan Masalah

Mata kuliah Desain Beton Bertulang adalah mata kuliah dasar di Program Studi yang berhubungan dengan teknik sipil. Fundamental yang kuat terhadap teori analisis and desain beton bertulang adalah sangat penting karena mayoritas bangunan gedung dan infrastruktur lainnya di Indonesia dibangun dengan menggunakan material beton. Peningkatan dalam proses belajar dan mengajar bisa ditingkatkan dengan aplikasi yang mendukung. Jadi saat ini ada kebutuhan akan alat bantu ajar untuk mata kuliah seperti Desain Beton Bertulang.

1.3 Ruang Lingkup Penelitian

Ruang lingkup dari penelitian ini adalah:

- a) Bahasa pemrograman yang digunakan adalah Python,
- b) Materi Beton Bertulang yang dibahas adalah analisis potongan penampang balok beton bertulang yang hanya menerima beban lentur/momen.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengembangkan aplikasi untuk membuat kurva kurvatur versus momen yang dapat digunakan untuk memperdalam pengertian teori analisis potongan penampang beton bertulang dengan menggunakan bahasa pemrograman Python.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah:

- a) Memudahkan dosen untuk menerangkan perilaku balok beton bertulang yang mengalami beban lentur,
- b) Membantu mahasiswa/i untuk lebih mendalami materi mengenai perilaku balok beton bertulang yang mengalami beban lentur.

1.6 Sistematika Penulisan

Laporan menjabarkan usaha dalam mengembangkan aplikasi yang dapat digunakan untuk mempelajari materi analisis potongan penampang beton bertulang yang mengalami beban lentur. Aplikasi ini dikembangkan dengan menggunakan bahasa pemrograman Python.

Laporan ini terdiri dari 5 Bab sebagai berikut:

Bab I menjelaskan latar belakang dari penelitian, indentifikasi masalah, ruang lingkup penelitian dan tujuan dan manfaat penelitian.

Bab II membahas studi Pustaka yang relevan dengan Bahasa pemrograman Python dan analisis lentur beton bertulang

Bab III membahas mengenai metode riset yang digunakan dalam penelitian ini.

Bab IV menampilkan dan mendiskusikan temuan dari penelitian ini.

Bab V memberikan kesimpulan dan rekomendasi.

BAB II

TINJAUAN PUSTAKA

2.1 Python

Python adalah bahasa pemrograman *interpreted* yang artinya program yang dibuat dalam bahasa pemrograman Python dapat dijalankan segera setelah file yang berisi *Python codes* selesai dibuat. Oleh karena itu, aplikasi yang dibuat dalam Python akan lebih mudah diperbaiki jika ada kesalahan dan jadi cepat selesai jika dibandingkan dengan program yang dibuat dengan bahasa pemrograman yang memerlukan *compiling*. Selain itu, Python juga dikenal sebagai bahasa pemrograman yang sangat mudah untuk dipelajari dan didukung oleh komunitas yang kuat dan modul-modul pendukung untuk berbagai bidang. Oleh karena itu Python menjadi bahasa pemrograman yang sangat populer baik bagi pemula maupun yang telah berpengalaman. Python juga mendukung berbagai gaya pemrograman yang meliputi pemrograman terstruktur dan berorientasi obyek (Srinath, 2017).

Seperti telah disinggung di atas, ada banyak modul-modul pendukung yang dapat mempermudah pengembangan suatu program. Untuk keperluan pembuatan grafik atau kurva, ada modul Matplotlib, yang merupakan kumpulan modul pendukung yang lengkap untuk pembuatan visualisasi statis, animasi dan interaktif dalam Python (Hunter, 2007). Sebagai contoh, kode sumber Python dan penggunaan modul Matplotlib yang cukup sederhana (Gambar 1.a akan menghasilkan grafik seperti pada Gambar 1.b. Jadi terlihat disini bahwa penggunaan modul dapat mempersingkat waktu pembuatan sebuah aplikasi. Modul lain yang akan dibahas disini adalah tkinter, modul yang ada di Python untuk membantu pembuatan *Graphical User Interface* (GUI).

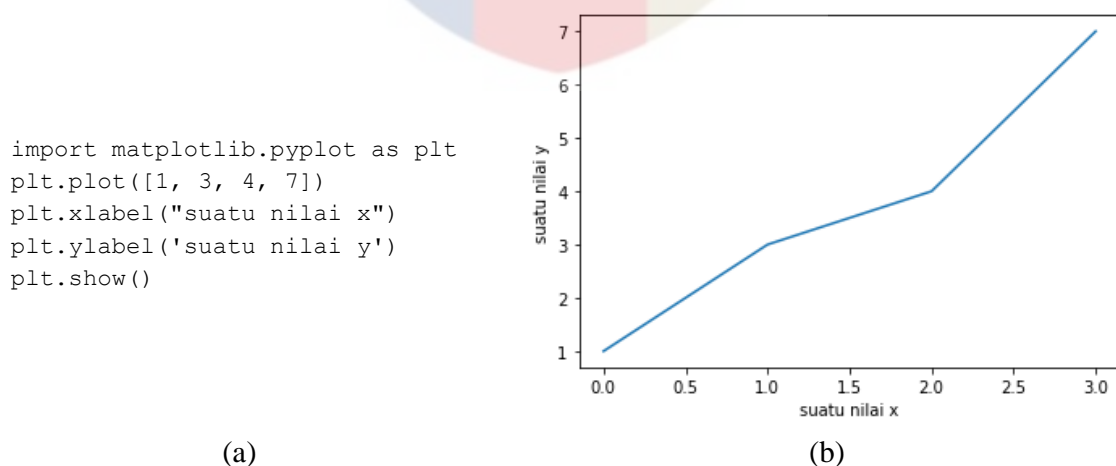


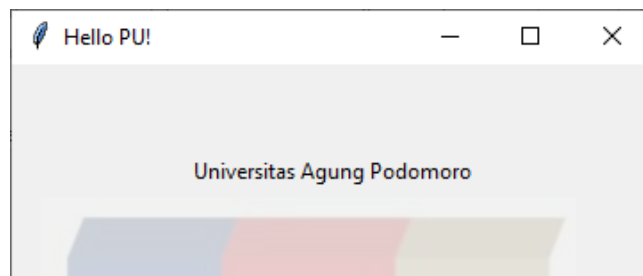
Figure 1. (a) Contoh kode sumber Matplotlib dalam Python dan (b) grafik yang dihasilkan

Aplikasi atau *software* yang dikembangkan di zaman sekarang wajib menggunakan GUI agar mudah digunakan dan menarik. Modul tkinter (*Tk interface*) adalah *interface* standar dalam Python ke *Tcl/Tk GUI toolkit* (Lundh, 1999 dan van Rossum, 2020). Baik Tk and tkinter tersedia untuk sistem operasi sejenis Unix, termasuk macOS, dan juga Windows. Aplikasi yang dibuat berdasarkan tkinter dapat dikatakan bisa digunakan pada setiap komputer walaupun dengan

sistem operasi berbeda. Gambar 2.a dan 2.b memperlihatkan penggunaan modul tkinter dalam pembuatan GUI sederhana.

```
import tkinter as tk
root = tk.Tk()
root.title('Hello PU!')
label = tk.Label(root, text='Universitas Agung Podomoro', padx=100, pady=50)
label.pack()
root.mainloop()
```

(a)



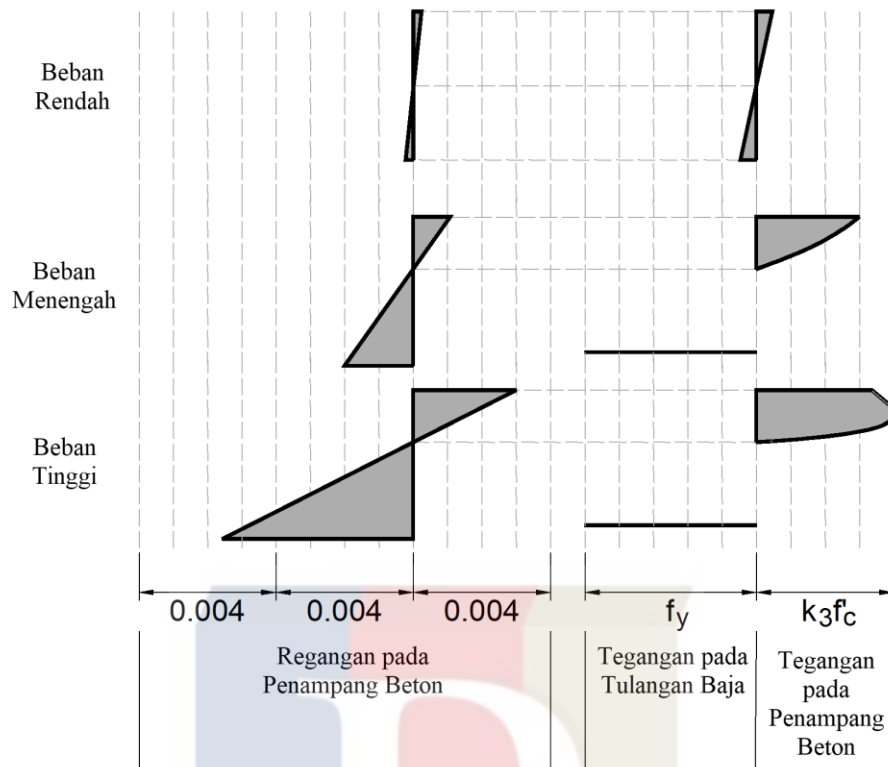
(b)

Figure 2. (a) Contoh kode sumber tkinter dalam Python dan (b) GUI yang dihasilkan

Terakhir yang akan dibahas mengenai Python adalah kesesuaiannya untuk digunakan dalam bidang sains dan rekayasa. Python sangat sesuai digunakan dalam bidang rekayasa seperti misalnya untuk perhitungan numerik (Kiusalaas, 2013). Ada modul `math` yang membantu dalam penggunaan fungsi-fungsi matematika seperti logaritma, akar dan lain sebagainya. Selain itu ada modul `numpy` yang dapat membantu dalam perhitungan yang berhubungan dengan vektor dan matriks.

2.2 Perilaku Balok Beton Bertulang yang Mengalami Beban Lentur atau Momen

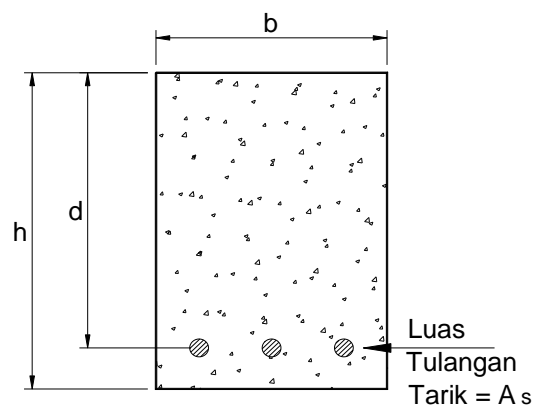
Salah satu prinsip dasar yang dipakai dalam menganalisis sebuah balok prisma yang mengalami lentur atau menerima beban momen adalah bahwa variasi regangan terjadi akan selalu linear di potongan penampangnya (Wight and MacGregor, 2012) seperti ditunjukkan pada Gambar 3. Terlihat disini bahwa regangan diasumsikan selalu linear di seluruh rentang pembebanan. Sedangkan untuk tegangan, pada saat beban rendah, maka variasinya linear, sedangkan pada saat beban momen lentur mulai membesar, maka bagian bawah dari penampang beton akan retak dan selanjutnya tegangan tarik akan ditanggung oleh tulangan baja.



Gambar 3 Bentuk tegangan dan regangan dari penampang beton pada level beban momen lenter yang berbeda

2.3 Hubungan momen dan kurvatur dari potongan penampang beton bertulang

Hubungan antara momen, kurvatur dan regangan secara kuantitatif dapat dijelaskan dengan meninjau sebuah potongan penampang yang mempunyai satu lapis tulangan seperti terlihat pada Gambar 4. Jika pada penampang tersebut bekerja momen seperti ditunjukkan pada Gambar 5, penampang tersebut kemungkinan akan mengalami 4 tahapan respons yang berbeda.

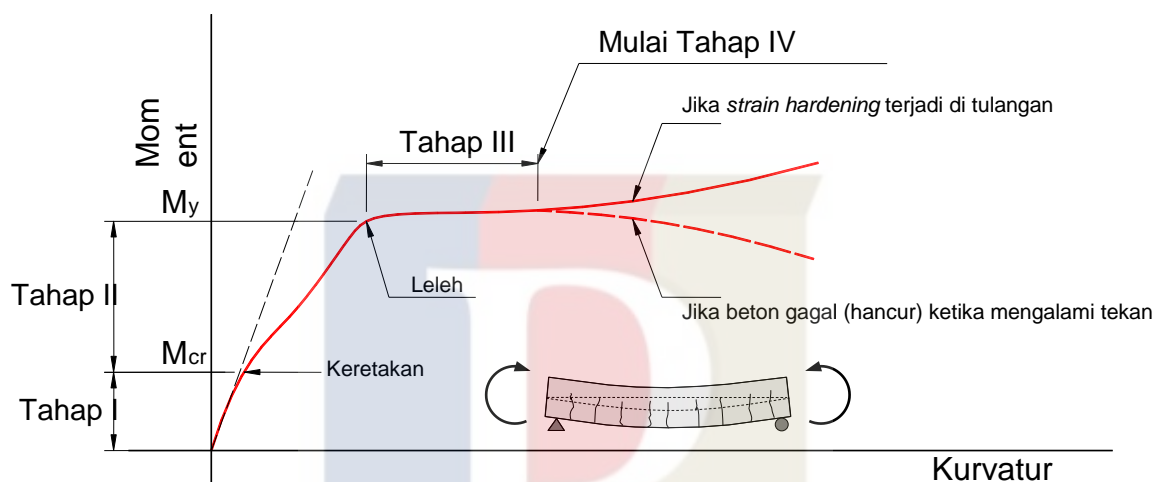


Gambar 3 Potongan penampang balok dengan 1 lapis tulangan

Pada tahap awal, respons dari penampang terhadap momen adalah linear (pertambahan momen

akan diikuti secara proportional dengan pertambahan deformasi/lendutan). Ketika momen yang bekerja mencapai nilai M_{cr} , maka beton di bagian bawah yang mengalami tegangan tarik akibat momen akan retak. Setelah momen melewati titik M_{cr} , laju perubahan deformasi bertambah dengan meningkatnya momen.

Selanjutnya, pada level momen leleh (*yield*), M_y , tulangan baja mencapai tegangan lelehnya. Pada kondisi ini, ada perubahan tiba-tiba pada kekakuan dari penampang. Deformasi akan terus terjadi dengan hampir tidak ada pertambahan momen. Tergantung dari luas tulangan yang diberikan, momen mungkin akan bertambah lagi pada deformasi yang besar seperti ditunjukkan, atau menurun seperti ditunjukkan pada Gambar 6



Gambar 5 Hubungan tipikal momen-kurvatur dari balok yang dibebani momen lentur

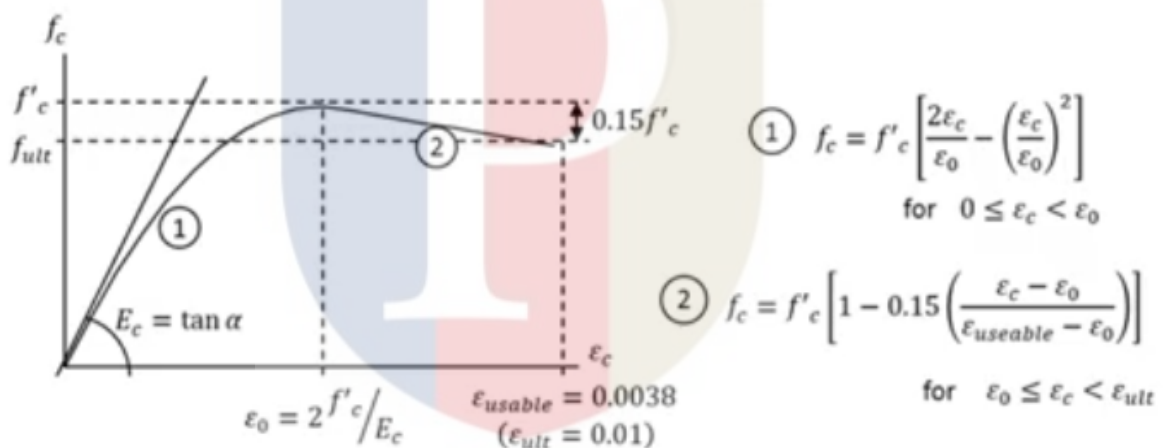
The hierarchy structure of an AHP model follows an inverted tree (Widianta et al., 2018).

BAB III METODE PENELITIAN

3.1 Prosedur Penelitian

Penelitian ini merupakan suatu pengembangan aplikasi mengenai perilaku balok beton bertulang yang mengalami beban momen lentur. Prosedur yang direncanakan agar penelitian ini dapat terlaksana adalah sebagai berikut:

1. Melakukan studi pustaka yang berhubungan dengan materi yang akan dibahas dalam penelitian ini seperti: Bahasa pemrograman Python dan perilaku balok beton bertulang terhadap beban lentur
2. Merumuskan masalah yang akan digunakan dalam pemrograman.
3. Membuat implementasi *Pseudocode* dari aplikasi yang akan dikembangkan
4. Menterjemahkan *Pseudocode* ke dalam kode sumber (source code) dalam bahasa pemrograman Python
5. Membahas hasil Program/Aplikasi yang dihasilkan

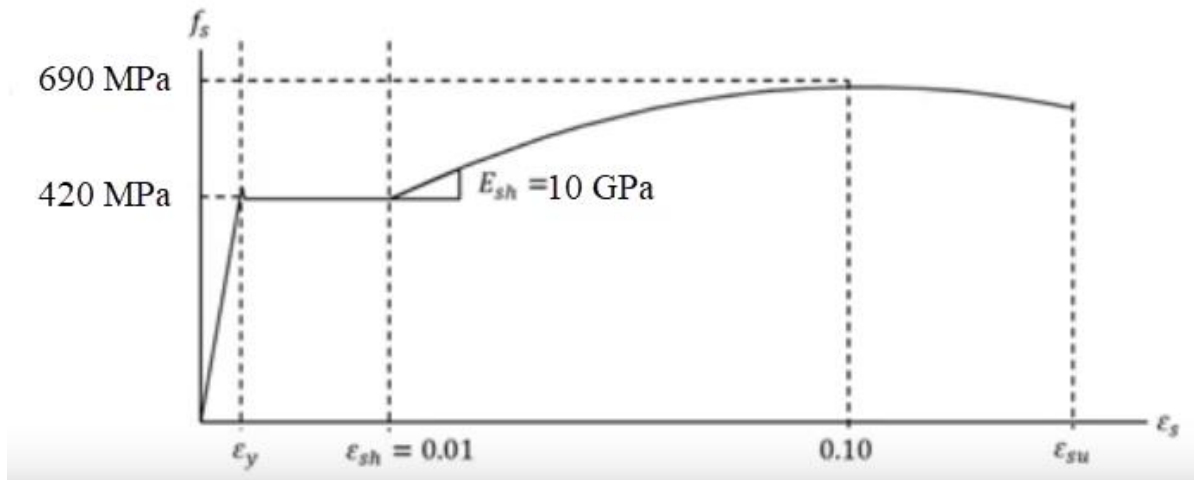


Gambar 6 Hubungan tegangan-regangan untuk beton (Hognestad)

3.2 Perumusan

Disini akan dibahas hubungan antara kurvatur, ϕ , dengan momen, M . Sebelum itu, perlu ditentukan model konstitutif dari material beton dan baja yang akan digunakan. Untuk beton, hubungan tegangan-regangan yang akan digunakan adalah yang dikembangkan oleh Hognestad seperti terlihat pada Gambar 6. Sedang untuk tulangan baja, hubungan tegangan-regangan yang dipakai adalah seperti yang diperlihatkan dalam Gambar 7. Untuk mendapatkan persamaan di daerah *strain hardening*, digunakan *cubic curve-fitting* dengan kondisi-kondisi batas berikut:

- $f_s(0,01) = 420 \text{ MPa}$
- $f_s(0,1) = 690 \text{ MPa}$
- $f'_s(0,01) = 10 \text{ GPa}$
- $f'_s(0,1) = 0$



Gambar 7 Hubungan tegangan-regangan untuk tulangan baja

Secara garis besar, kurvatur dan momen dari potongan penampang beton akan ditentukan pada kondisi-kondisi berikut:

- a) ϕ sebelum beton retak
Digunakan properti penampang transformasi sebelum retak. Lengan momen ditentukan menggunakan keseimbangan statis momen penampang.
- b) ϕ setelah beton retak
Digunakan properti penampang transformasi setelah retak. Lengan momen ditentukan menggunakan keseimbangan statis momen penampang.
- c) ϕ pada saat $f_c = 0.45f_c'$ (asumsi batas beton masih berperilaku elastis linear)
Disini diasumsikan beton dan tulangan baja masih elastik. Ini berarti sumbu netral tidak berubah dari yang didapat di c). Lengan momen ditentukan menggunakan keseimbangan statis momen penampang.
- d) $\phi = \phi_y$ (tulangan baja meleleh)
Kondisi ini terjadi pada saat tulangan baja mulai leleh ($\epsilon_s = \epsilon_y$). Disini kita tidak bisa menggunakan properti potongan penampang transformasi karena perilaku baja sudah tidak lagi linear. Regangan pada lapisan atas beton masih di bawah regangan ultimit. Lengan momen ditentukan menggunakan keseimbangan gaya-gaya yang bekerja.
- e) ϕ pada saat $\epsilon_c = \epsilon_{c0}$
Pada kondisi ini, tegangan beton mencapai tegangan ultimit. Untuk baja tulangan, kondisinya bisa diasumsikan belum mencapai *strain hardening* atau sudah, yang kemudian harus dicek kebenaran asumsinya. Lengan momen ditentukan menggunakan keseimbangan gaya-gaya yang bekerja.
- f) ϕ pada $\epsilon_c = \epsilon_{usable}$
Regangan beton mencapai kondisi batas maksimum yang umumnya merupakan nilai batas. Lengan momen ditentukan menggunakan keseimbangan gaya-gaya yang bekerja.
- g) ϕ pada $\epsilon_s = \epsilon_{strain\ hardening}$
Regangan pada baja diasumsikan mencapai strain hardeing, lengan momen ditentukan menggunakan keseimbangan gaya-gaya yang bekerja.
- h) ϕ pada $\epsilon_c = \epsilon_{ultimate}$
Disini kita mengasumsikan regangan beton maksimum dapat mencapai 0,01. Kondisi ini dapat tercapai untuk beton yang terkekang dengan baik. Lengan momen ditentukan menggunakan keseimbangan gaya-gaya yang bekerja.

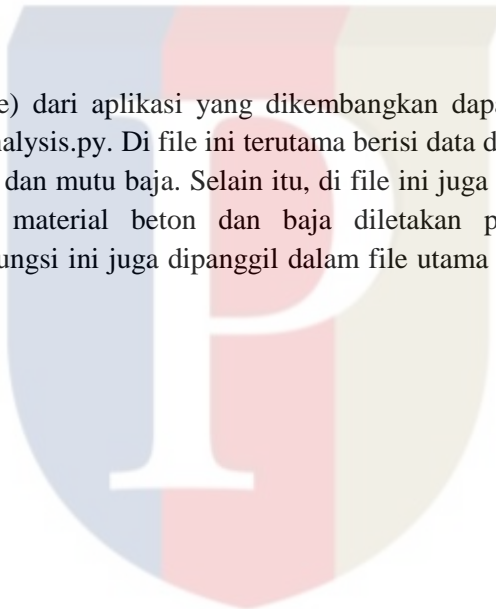
3.3 Implementasi dalam bentuk *Pseudocode*

Implementasi dalam perumusan dilakukan dalam bentuk *Pseudocode* sebagai berikut:

- Menyediakan data-data yang dibutuhkan oleh program, seperti dimensi dari penampang, mutu beton dan mutu baja tulangan yang akan digunakan
- Membuat fungsi untuk menentukan tegangan dari regangan yang dimasukkan. Fungsi dibuat untuk material beton dan juga baja
- Menentukan nilai kurvatur dan momen untuk kondisi-kondisi yang didefinisikan dalam perumusan
- Menyusun hasil-hasil kurvatur dan momen untuk dimasukkan ke dalam Matplotlib
- Mempersiapkan GUI untuk menampilkan hasil

3.4 Pemrograman

Kode sumber (source code) dari aplikasi yang dikembangkan dapat dilihat pada Lampiran. File utama diberi nama `conc_analysis.py`. Di file ini terutama berisi data dimensi penampang beton, mutu beton, berat volume beton, dan mutu baja. Selain itu, di file ini juga berisi kode sumber plotting dan GUI. Fungsi-fungsi dari material beton dan baja diletakan pada file lain, dengan nama `concrete_bem.py`. Fungsi-fungsi ini juga dipanggil dalam file utama untuk memproses perhitungan-perhitungan.

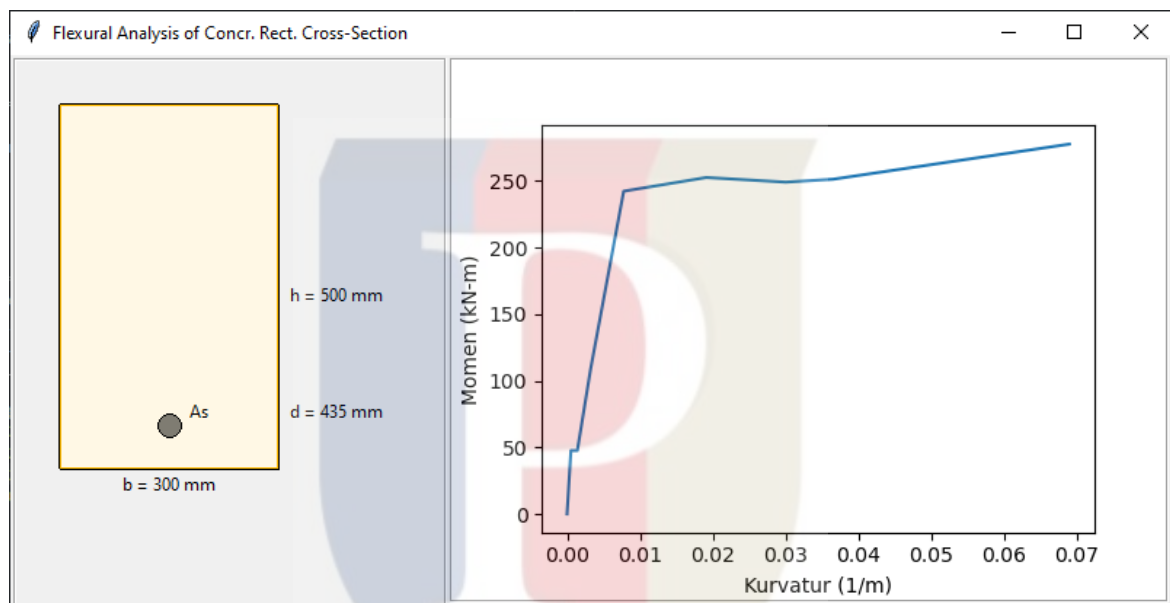


BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil

Hasil penelitian adalah aplikasi untuk melakukan penggambaran perilaku balok beton bertulang jika mengalami beban lentur. Dengan memasukan data penampang balok beton bertulang dengan tinggi 500 mm dan lebar 300 mm, data mutu beton $f'c = 28$ MPa dan mutu baja tulangan $f_y = 420$ MPa, maka didapatkan hubungan momen-kurvatur seperti terlihat pada Gambar 8.



Gambar 8 Interface aplikasi dan hubungan kurvatur-momen yang dihasilkan

Pembahasan

Grafik kurvatur-momen yang dihasilkan dari aplikasi berhasil menangkap mekanisme yang terjadi jika balok beton bertulang dibebani lentur hingga mendekati batas runtuhnya. Proses retak, yang diikuti pembentukan keseimbangan penampang dapat terlihat dengan jelas di grafik. Kemudian pada saat tulangan baja mulai leleh, tegangan beton mencapai ultimit, baja mencapai kondisi strain hardening, regangan beton mencapai batas pakai, dan akhirnya regangan beton diasumsikan mencapai kondisi ultimit. Karena grafik ini dihasilkan dari perhitungan yang mengikuti kaidah-kaidah yang berlaku, maka ini dapat meningkatkan pemahaman menjadi bahan pembelajaran karena ini setara dengan mempelajari hasil eksperimen

BAB V

SIMPULAN DAN SARAN

Penelitian yang mengembangkan aplikasi untuk membuat kurva kurvatur versus momen yang dapat digunakan untuk memperdalam pengertian teori analisis potongan penampang beton bertulang dengan menggunakan bahasa pemrograman Python telah berhasil dicapai. Aplikasi dapat berjalan sesuai dengan yang direncanakan. Semua perilaku seperti retak, leleh, ulimit pada potongan penampang balok beton bertulang yang mengalami beban lentur berhasil disimulasikan oleh aplikasi.

5.2 SARAN

Penelitian lebih lanjut dapat dikembangkan dengan mengambarkan diagram regangan dan tegangan pada setiap kondisi disetiap tahap pembebanan.



DAFTAR PUSTAKA

- Hunter, J.D. (2007). Matplotlib: A 2D Graphics Environment. in *Computing in Science & Engineering*. 9(3): 90-95. doi: 10.1109/MCSE.2007.55.
- Kiusalaas, J. (2013). *Numerical methods in engineering with Python 3*. Cambridge: Cambridge University Press.
- Lundh, F. (1999). An introduction to tkinter. URL: www.Pythonware.Com/Library/Tkinter/Introduction/Index.htm.
- Srinath, K.R. (2017). Python – The Fastest Growing Programming Language. *International Research Journal of Engineering and Technology*. 4(12): 354-357.
- Van Rossum, G. (2020). *The Python Library Reference*, release 3.8.2. Python Software Foundation.
- Wight, J.K. and MacGregor, J.G. (2012) *Reinforced Concrete: Mechanics and Design*. Pearson Education, Inc., Upper Saddle River, New Jersey.



Lampiran 1. Alokasi Waktu dan Uraian Tugas

Penelitian ini bekerjasama dengan seorang dosen Universitas Trisakti dan seorang praktisi di industri konstruksi.

No.	Nama / NIDN	Instansi Asal	Bidang Ilmu	Alokasi Waktu (jam/minggu)	Uraian Tugas
1	Suwito, ST., PhD. (0310036702)	Universitas Agung Podomoro	Manajemen & Rekayasa Konstruksi	8	Studi literatur, pengumpulan data, analisis data, penajaman diskusi, laporan

Lampiran 2. Realisasi Anggaran

Penelitian ini memanfaatkan peralatan yang sudah ada dan menggunakan *free development software* sehingga tidak membutuhkan anggaran penelitian.

Lampiran 3. Publikasi Jurnal

Hasil penelitian ini rencananya tidak dipublikasikan.

Lampiran 4. Source Code (conc_analysis.py)

```
import concrete_beam as cb
from tkinter import *
from tkinter import ttk
import matplotlib.pyplot as plt
from matplotlib.figure import Figure
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg

class Window(Tk):
    def __init__(self):
        super().__init__()
        self.title("Flexural Analysis of Concr. Rect. Cross-Section")

# label = ttk.Label(self, text="Concrete Beam Analysis")
# label.pack(fill=BOTH, expand=1, padx=100, pady=50)
if __name__ == "__main__":
    b_mm = 300
    h_mm = 500
    d_mm = 435
    f_prime_c_mpa = 28
    density_kncm = 23.5
    fy_mpa = 420

    rect_conc = cb.RectBeam(b_mm, h_mm, f_prime_c_mpa, density_kncm)
    reinf_bar = cb.ReinfBar(fy_mpa)

    as_min_mm2 = rect_conc.as_min(d_mm, fy_mpa)
    print("As min = ", as_min_mm2)
    as_max_mm2 = rect_conc.as_max(d_mm, fy_mpa)
    print("As max = ", as_max_mm2)

    as_mm2 = as_min_mm2+(as_max_mm2-as_min_mm2)*2.28/4
    print("As = ", as_mm2)
    m_crack = rect_conc.moment_crack_nmm(as_mm2, d_mm, reinf_bar.es_mpa)
    print("m_crack =", m_crack/1000000)
    curv_cr_1 = rect_conc.curv_cr_1(m_crack, as_mm2, d_mm, reinf_bar.es_mpa)
    curv_cr_2, cna2 = rect_conc.curv_cr_2(m_crack, as_mm2, d_mm, reinf_bar.es_mpa)
    curv_cr_3 = rect_conc.eps_c45/cna2
    m_crack3 = curv_cr_3 * m_crack / curv_cr_2
    cna3 = 165
    ec_t4 = reinf_bar.eps*(cna3/(d_mm-cna3))
    curv_cr_4 = ec_t4/cna3
    y_mm = 107
    m_y = as_mm2*reinf_bar.fy_mpa * (d_mm - cna3 + y_mm)
    cna4 = 113
    curv_cr_5 = rect_conc.eps_0/cna4
    y4_mm = 71
    m_eps0 = as_mm2*reinf_bar.fy_mpa * (d_mm - cna4 + y4_mm)

    cna5= 102
    eps_ct = reinf_bar.eps_s_sh*(cna5/(d_mm - cna5))
    curv_cr_6 = eps_ct/cna5
```

```

y5_mm = 61
t_s_n = 6.325e5
m_sh = t_s_n * (d_mm - cna5 + y5_mm)

cna6 = 104
curv_cr_7 = rect_conc.eps_u/cna6
y6_mm = 60
m_epsu = as_mm2*reinf_bar.fy_mpa * (d_mm - cna6 + y6_mm)

cna7 = 145
curv_cr_8 = rect_conc.eps_ult/cna7
y7_mm=65
t_s_n = 7.824e5
m_ult = t_s_n * (d_mm - cna7 + y7_mm)

eps_comp2 = cna2 * curv_cr_2
eps_tens2 = (d_mm-cna2) * curv_cr_2
eps_comp3 = cna2 * curv_cr_3
eps_tens3 = (d_mm-cna2) * curv_cr_2

print("eps_c2, eps_t2", eps_comp2, eps_tens2)

print("eps_c3, eps_t3", eps_comp3, eps_tens3)

# curv_cr_2 = curv_cr_1
m_inc = (0, m_crack/1e6, m_crack/1e6, m_crack3/1e6, m_y/1e6, m_eps0/1e6,
        m_sh/1e6, m_epsu/1e6, m_ult/1e6)
curv_inc = (0, curv_cr_1*1e3, curv_cr_2*1e3, curv_cr_3*1e3, curv_cr_4*1e3,
           curv_cr_5*1e3, curv_cr_6*1e3, curv_cr_7*1e3, curv_cr_8*1e3)
plt.plot(curv_inc, m_inc)
plt.xlabel("Kurvatur (1/m)")
plt.ylabel("Momen (kN-m)")
# plt.show()
plt.show()

window = Window()

frame = ttk.Frame(master=window, width = 800, height = 380)
frame.pack()
frame_left = ttk.Frame(master=frame, width = 296, height = 376,
                      borderwidth = 2, relief = GROOVE
                      )
frame_left.place(x = 2, y = 2)

frame_right = ttk.Frame(master=frame, width = 396, height = 376,
                       borderwidth = 2, relief = GROOVE)

frame_right.place(x = 302, y = 2)

```

```

# frame_bot.place(x = 2, y = 452)
canv1 = Canvas(frame_left, width = 290, height = 370)
# canv1.place(x = 10, y = 10)
coord = 30, 30, 180, 30, 180, 280, 30, 280, 30, 30
line = canv1.create_line(coord, fill="black", width=2)
canv1.pack(fill = BOTH)

# the figure that will contain the plot
fig = Figure(figsize = (4.9, 3.7), dpi = 100)

# adding the subplot
plot1 = fig.add_subplot(111)

# plotting the graph
plot1.plot(curv_inc, m_inc)
plot1.set_ylabel("Momen (kN-m)")
plot1.set_xlabel("Kurvatur (1/m)")

# creating the Tkinter canvas
# containing the Matplotlib figure
canvas = FigureCanvasTkAgg(fig, master = frame_right)

canvas.draw()

# placing the canvas on the Tkinter window
canvas.get_tk_widget().pack()

window.mainloop()

```

Lampiran 5. Source Code (concrete_beam.py)

```

import math as m

class ReinfBar:
    def __init__(self, fy_mpa):
        self.bar = {0: [0, 0], 8: [8, 50], 10: [10, 79], 13: [13, 132], 16:
            [16, 201], 19: [19, 284],
            22: [22, 380], 25: [25, 491], 28: [28, 616], 32: [32, 804],
            36: [36, 1018], 40: [40, 1257]}
        self.fy_mpa = float(fy_mpa)
        self.fy_gpa = self.fy_mpa / 1000
        self.es_mpa = 200000
        self.es_gpa = self.es_mpa / 1000
        self.eps = self.fy_mpa/self.es_mpa
        self.eps_s_sh = 0.01

class RectBeam:
    def __init__(self, b_mm, h_mm, f_prime_c_mpa, density_kncm):

```

```

self.b_mm = float(b_mm)
self.h_mm = float(h_mm)
self.f_prime_c_mpa = float(f_prime_c_mpa)

self.ig_mm4 = (self.b_mm * self.h_mm ** 3) / 12

if 14 <= density_kncm <= 25:
    self.ec_mpa = density_kncm ** 1.5 * 43 * self.f_prime_c_mpa ** \
        (1 / 2.0)
else:
    self.ec_mpa = 'Density of Concrete must be between 14 and 25 kN/m3'

self.weight_nlm = (self.b_mm * self.h_mm) * (1 / 1000) * density_kncm
self.weight_knlm = self.weight_nlm / 1000.0
self.fr_mpa = 0.62 * self.f_prime_c_mpa ** (1 / 2.0)
self.m_crack_nmm = (self.fr_mpa * self.ig_mm4) / (self.h_mm / 2)
self.m_crack_nm = self.m_crack_nmm / 1000.0
self.m_crack_knm = self.m_crack_nm / 1000.0

if self.f_prime_c_mpa <= 28:
    self.beta1 = 0.85
elif self.f_prime_c_mpa <= 56:
    self.beta1 = 0.85 - ((0.05 * (self.f_prime_c_mpa - 28)) / 7)
else:
    self.beta1 = 0.65

self.f_prime_c45_mpa = 0.45 * self.f_prime_c_mpa
self.eps_c45 = self.f_prime_c45_mpa / self.ec_mpa
self.eps_0 = 2 * self.f_prime_c_mpa / self.ec_mpa
self.eps_u = 0.0038
self.eps_ult = 0.01

# section vertex coordinates
self.section_x_coords_mm = [0, self.b_mm, self.b_mm, 0, 0]
self.section_y_coords_mm = [0, 0, self.h_mm, self.h_mm, 0]

def as_min(self, d_mm, fy_mpa):
    return max((0.25 * self.f_prime_c_mpa ** (1 / 2.0) * self.b_mm * d_mm)
        / fy_mpa, (1.4 * self.b_mm * d_mm) / fy_mpa)

def as_max(self, d_mm, fy_mpa):
    return (0.3188 * self.beta1 * self.f_prime_c_mpa * self.b_mm * d_mm /
        fy_mpa)

def strain_compatibility_steel(self, bars_as, bars_d, c_mm, fy_mpa, es_mpa):
    steel_strain = 0
    steel_stress = 0
    steel_tension_force_kn = 0
    i = 0
    steel_strain = 0.003 * ((bars_d / c_mm) - 1)
    fs = steel_strain * es_mpa

```

```

if fs < -1 * fy_mpa or fs > fy_mpa:
    steel_stress = ((fs / abs(fs)) * fy_mpa)
    steel_tension_force_n = ((fs / abs(fs)) * steel_stress) * bars_as
else:
    steel_stress = fs
    steel_tension_force_n = fs * bars_as

return steel_strain, steel_stress, steel_tension_force_n

def strain_compatibility_concrete(self, c_mm):
    a = self.beta1 * c_mm
    compression_block_mm2 = self.b_mm * a
    concrete_compression_force_n = 0.85 * self.f_prime_c_mpa * compression_block_mm2
    compression_block_cg_top = a / 2

    return concrete_compression_force_n, compression_block_cg_top
def fc(eps_c):
    if eps_c <= self.eps_0:
        return self.f_prime_c_mpa*(2*eps_c/self.eps_0-(eps_c/self.eps_0)**2)
    elif eps_0 < eps_c <= 0.003:
        return 0.85*self.f_prime_c_mpa+self.f_prime_c_mpa(0.45-150*eps_c)

def find_pna(self, bars_as, bars_d, fy_mpa, es_mpa):
    a = 0
    b = bars_d
    c = 0
    pna = 0

    loop_max = 10000
    tol = 0.0000000000001
    loop = 0

    while loop < loop_max:
        c = (a + b) / 2
        strain_c, stress_c, tension_c = self.strain_compatibility_steel
        (bars_as, bars_d, c, fy_mpa, es_mpa)
        compression_c, compression_cg = self.strain_compatibility_concrete(c)

        if compression_c == tension_c or (b - a) / 2 <= tol:
            pna = c
            loop = loop_max
        elif compression_c > tension_c:
            b = c
        else:
            a = c
        loop += 1
    return pna

def moment_capacity_nmm(self, bars_as, bars_d, c_mm, fy_mpa, es_mpa):
    a = self.beta1 * c_mm

```

```

compression_block_mm2 = self.b_mm * a
concrete_compression_force_n = 0.85 * self.f_prime_c_mpa * compression_block_mm2

steel_strain = 0
steel_stress = 0
steel_tension_force_n = 0
steel_moment_component_nmm = 0

steel_strain = (0.003 * ((bars_d / c_mm) - 1))
fs = steel_strain * es_mpa
if fs < -1 * fy_mpa or fs > fy_mpa:
    steel_stress = ((fs / abs(fs)) * fy_mpa)
    steel_tension_force_n = ((fs / abs(fs)) * fy_mpa) * bars_as
    steel_moment_component_nmm += steel_tension_force_n * (bars_d - c_mm)
else:
    steel_stress = fs
    steel_tension_force_n = fs * bars_as
    steel_moment_component_nmm += steel_tension_force_n * (bars_d - c_mm)

if max(steel_strain) < 0.004:
    return 'Section not Tension Controlled'
else:
    if max(steel_strain) >= 0.005:
        phi = 0.9
    else:
        phi = 0.65 + ((max(steel_strain) - 0.002) * (250 / 3))

concrete_moment_arm_mm = c_mm - (a / 2)
concrete_moment_component_nmm = concrete_compression_force_n * \
    concrete_moment_arm_mm
nominal_moment_nmm = concrete_moment_component_nmm + \
    steel_moment_component_nmm

return phi, nominal_moment_nmm, phi * nominal_moment_nmm

```

```

def cracked_moment_of_inertia_mm4(self, bars_as, bars_d, es_mpa):

```

```

    n = es_mpa / self.ec_mpa

    a = 0
    b = bars_d
    c = 0
    mna = 0
    na = 0

    loop_max = 10000
    tol = 0.00000000000001
    loop = 0
    na1 = -n*bars_as/self.b_mm+((n*bars_as/self.b_mm)**2+

```

```

                2*n*bars_as*bars_d/self.b_mm)**0.5
while loop < loop_max:
    c = (a + b) / 2

    mna = self.b_mm * c * (c / 2)

    if c < bars_d:
        mna = mna - (n * bars_as * (bars_d - c))
    else:
        mna = mna + ((n - 1) * bars_as * (c - bars_d))

    if mna == 0 or (b - a) / 2 <= tol:
        na = c
        loop = loop_max
    elif mna > 1:
        b = c
    else:
        a = c
    loop += 1
# print na

i_crack_mm4 = (self.b_mm * na ** 3) / 3

if na < bars_d:
    i_crack_mm4 = i_crack_mm4 + n * bars_as * (bars_d - na) ** 2
else:
    i_crack_mm4 = i_crack_mm4 + (n - 1) * bars_as * (na - bars_d) ** 2

print("na = ", na)
print("na1 = ", na1)
return na, i_crack_mm4

def uncracked_moment_of_inertia_mm4(self, bars_as, bars_d, es_mpa):
    n = es_mpa / self.ec_mpa
    y_na = ((self.h_mm / 2) * self.b_mm * self.h_mm + bars_d * (n - 1) * bars_as) \
        / (self.b_mm * self.h_mm + (n - 1) * bars_as)
    i_g = (1 / 12) * self.b_mm * self.h_mm ** 3 + (self.h_mm / 2 - y_na) ** 2 \
        * self.b_mm * self.h_mm + (bars_d - y_na) ** 2 * (n - 1) * bars_as
    return y_na, i_g

def moment_crack_nmm(self, bars_as, bars_d, es_mpa):
    y_na, i_g = self.uncracked_moment_of_inertia_mm4(bars_as, bars_d, es_mpa)
    return self.fr_mpa * i_g / (self.h_mm - y_na)

def curv_cr_1(self, m_nmm, bars_as, bars_d, es_mpa):
    y_na, i_g = self.uncracked_moment_of_inertia_mm4(bars_as, bars_d, es_mpa)
    curv = m_nmm / (self.ec_mpa * i_g)
    return curv

def curv_cr_2(self, m_nmm, bars_as, bars_d, es_mpa):
    na, i_g = self.cracked_moment_of_inertia_mm4(bars_as, bars_d, es_mpa)
    curv = m_nmm / (self.ec_mpa * i_g)
    return curv, na

```