LAPORAN FINAL RESEARCH-BASED LEARNING

Pemrograman dalam Sains SK 5002 Math Games for Kids

20921003 Winda Wijaya 20921004 Mohammad Rizka Fadhli

14 November 2021

## Abstrak

Anak-anak generasi *alpha* (kelahiran 2011 - 2025) adalah generasi yang akrab dengan dunia *digital* dan teknologi (*digital natives*) (McCrindle, 2020). Mereka terlahir dengan berbagai teknologi yang sudah maju beserta segala kemudahannya. Pandemi yang datang secara mendadak mau tidak mau harus mengubah cara orang tua dan guru untuk mengajarkan materi pelajaran seperti berhitung. Program *math gamse for kids* menawarkan kemudahan baik bagi orang tua, guru, dan anak dalam hal memberikan soal latihan sekaligus melakukan evaluasi secara otomatis. Program ini dibuat menggunakan bahasa *Python* yang sudah populer dalam dunia *programming* dan edukasi.

# PENDAHULUAN

## Latar Belakang

Anak kelas 3 - 4 SD diharapkan sudah bisa mengerjakan soal penjumlahan, pengurangan, perkalian, dan pembagian yang sederhana. Untuk itu, mereka perlu mendapatkan latihan soal yang cukup. Sejak memasuki masa pandemi **Covid 19**, metode pendidikan berubah ke arah latihan mandiri, dengan kendala-kendala yang sering ditemui sebagai berikut:

* Buku pelajaran dan latihan soal yang terbatas,
* Keterbatasan guru atau orang tua dalam mengecek pekerjaan siswa,
* Tingginya distraksi dari aktivitas lain seperti *video game* terutama untuk generasi *alpha* yang sangat fasih terhadap teknologi.

Oleh karena itu, muncul ide untuk membuat suatu program yang bisa membuat soal latihan berhitung secara *random* dan otomatis dengan tingkat kesulitan soal yang dapat disesuaikan. Sehingga akan tercipta soal latihan tidak terbatas banyaknya. Selain itu diharapkan anak menjadi senang karena akan terlihat seperti bermain *game* di komputer atau *gadget*.

## Tujuan

Membuat program yang dapat membuat soal latihan berhitung dengan sifat-sifat sebagai berikut:

1. Soal dibuat dengan *random*,
2. Soal dibuat secara otomatis,
3. Tingkat kesulitan soal dapat disesuaikan,
4. Proses evaluasi soal dilakukan secara otomatis.

## Limitasi Penelitian

Program ini ditujukan kepada anak kelas 3 - 4 SD. Oleh karena itu, setiap bilangan yang ada di soal merupakan **bilangan** ***integer*** (bulat).

## Metode

Program akan dibuat menggunakan bahasa pemrograman *Python* versi 3 dengan memanfaatkan beberapa *libraries* tambahan.

## Target *Output* Penelitian

Penelitian ini diharapkan akan menghasilkan suatu program yang dapat dijalankan di komputer atau *gadget* berbasis *Python* sebagai *interpreter*-nya. Untuk itu, kami membuat dua tahapan pengerjaan, yakni:

* **TAHAP I** berupa program dalam bentuk *command line interface* (**CLI**). Program ini memiliki semua minimal fitur yang dibutuhkan seperti:
  + *Generate random* soal,
  + Evaluasi jawaban anak,
  + *Plot* evaluasi dalam grafik,
  + Penyimpanan rekap jawaban anak dalam suatu file berformat .csv.
* **TAHAP II** berupa program dalam bentuk *graphical user interface* (**GUI**).
  + Fitur yang dimiliki:
    - *Generate random* soal,
    - Evaluasi jawaban anak,
    - *Sound effects* saat jawaban benar atau salah.
  + Fitur yang belum dimiliki:
    - *Plot* evaluasi dalam grafik,
    - Penyimpanan rekap jawaban anak dalam suatu file berformat .csv.

# RINGKASAN STUDI PUSTAKA

## *Games* Edukasi Anak

Sampai dengan saat ini, sudah banyak *games* anak yang edukatif dan melatih kemampuan matematika seperti:

sumber: understood.org

|  |  |  |
| --- | --- | --- |
| Jenis games | Contoh games | Kemampuan yang Dilatih |
| Mencocokkan | Go Fish & Domino | Memori |
| Misteri | Tebak-tebakan | Logika |
| Strategi spasial | Catur & Battleship | Memori, logika, & spasial |
| Strategi angka | Uno & Mancala | Aritmatika & identifikasi angka |
| Manajemen sumber daya | Monopoli | Aritmatika & manajemen |

Seiring dengan waktu, permainan berbasis komputasi mulai dibuat dan berkembang dengan sangat pesat. Perkembangan ini sedikit banyak disebabkan oleh semakin banyaknya bahasa pemograman dengan berbagai kemampuan baru yang ditawarkan untuk membangun permainan komputer yang lebih menarik.

Bermain *game* matematika mendukung berpikir strategis, menyelesaikan masalah, dan meningkatkan kelancaran dalam mengerjakan soal[[1]](#footnote-27). Di Indonesia sendiri, telah diimplementasikan **Ujian Nasional Berbasis Komputer** (**UNBK**) sebagai salah satu syarat kelulusan. Siswa tidak hanya perlu untuk memahami pelajaran, tetapi juga harus terbiasa mengerjakan soal menggunakan komputer[[2]](#footnote-29). Oleh karena itu prilaku terbiasa mengerjakan soal di komputer perlu dilatih karena akan sangat membantu kelancaran dan ketenangan siswa dalam menyelesaikan ujian.

## Python

Bahasa yang digunakan untuk membuat *game* ini adalah *Python*. Disebut demikian karena pembuat bahasa tersebut adalah penggemar serial televisi Inggris ***Monty Python***. *Python* populer dalam pengembangan *game*, selain pengembangan aplikasi hingga situs *web*. Bahkan **NASA** dan **Google** sangat bergantung pada *Python* (McGuggan, 2007). Game populer seperti *Snakes*, *Flappy Bird*, *Tetris*, *Battlefield 2*, *The Sims 4*, *World of Tanks*, *Disney Toontown Online* dan masih banyak lagi menggunakan bahasa *Python* untuk membuatnya[[3]](#footnote-31).

Beberapa *libraries* lain yang digunakan pada penyusunan program ini antara lain:

1. random: digunakan untuk mendapatkan *function* randint(). *Function* ini berguna untuk *generate* bilangan *integer*.
2. datetime: digunakan untuk mendapatkan function datetime.now(). *Function* ini berguna untuk mendapatkan waktu saat suatu perintah di-eksekusi. Kelak kita akan menghitung durasi anak menjawab sebuah soal.
3. matplotlib: digunakan untuk membuat dan menyimpan grafik rekap jawaban anak.
4. numpy: digunakan untuk menghitung dan merekap evaluasi jawaban anak.
5. pygames: digunakan untuk membuat **GUI** *games*.

## Prinsip Komputasi

Beberapa prinsip komputasi di Python yang digunakan antara lain:

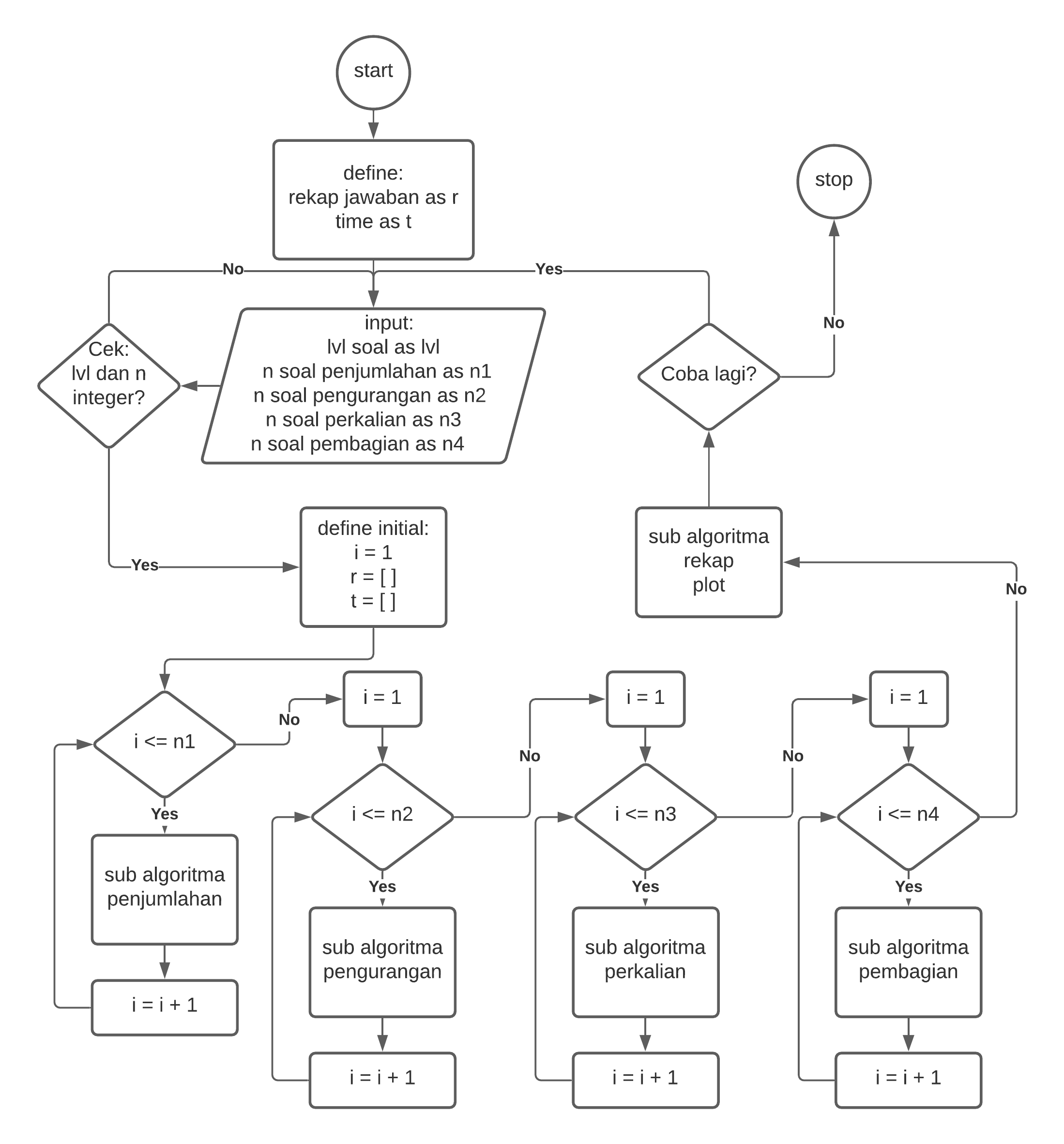
* *Function*
  + Digunakan untuk memecah keseluruhan algoritma menjadi beberapa sub modul (sub algoritma) kecil sehingga lebih *manageable*.
* *Looping*
  + for() – digunakan untuk generate n-buah soal.
  + while() – digunakan untuk memastikan bahwa *input* yang dimasukkan berupa *integer*. Jika tidak, program akan meminta *user* untuk memasukkan *input* kembali.
* *Conditional*
  + Untuk mengecek jawaban anak.
* *Data storage*
  + Menyimpan rekap jawaban.
* Visualisasi data

# HASIL PENELITIAN

## Program Versi CLI

### *Flowchart Utama*

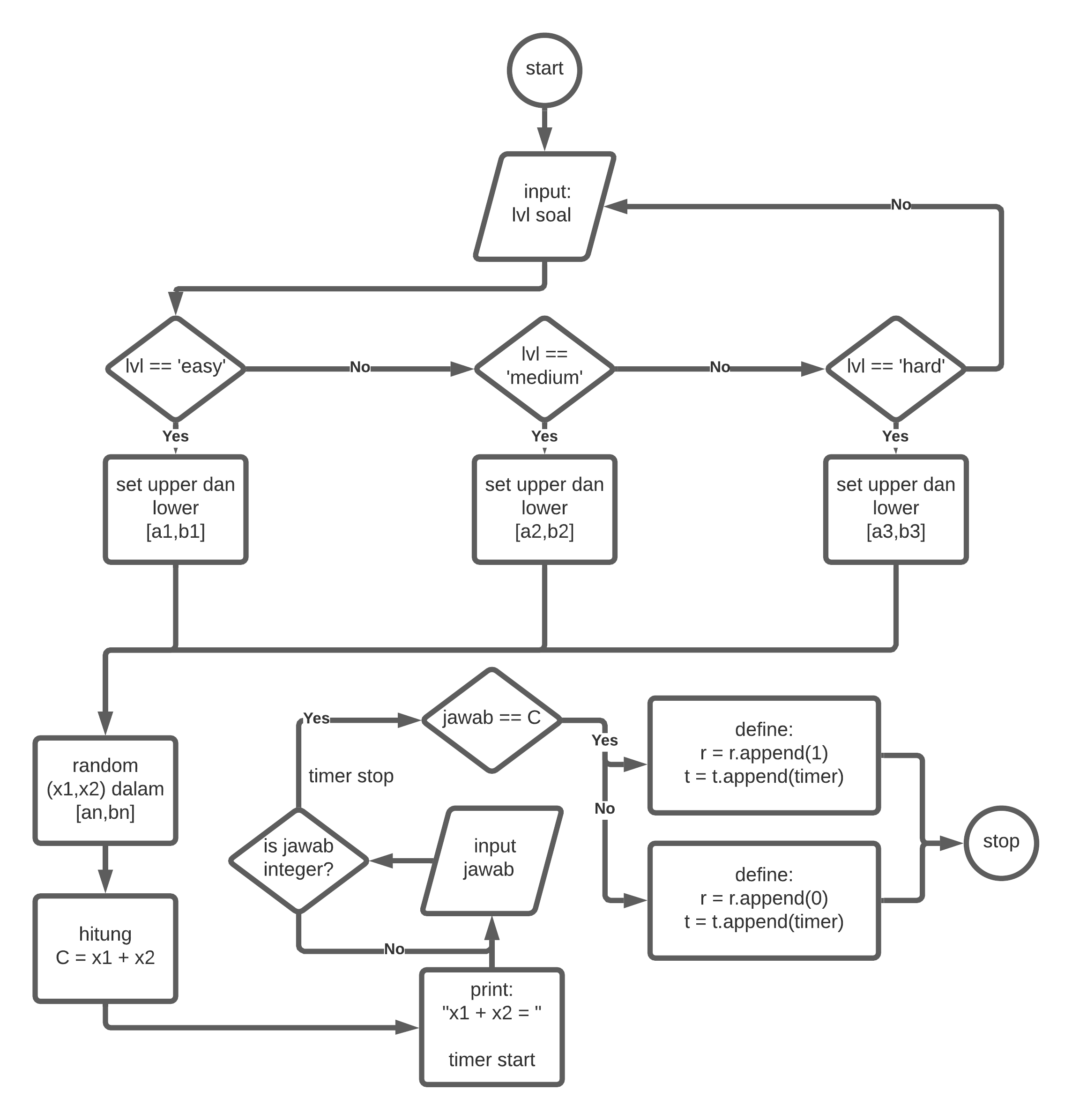
Berikut adalah *flowchart* dari program ini:



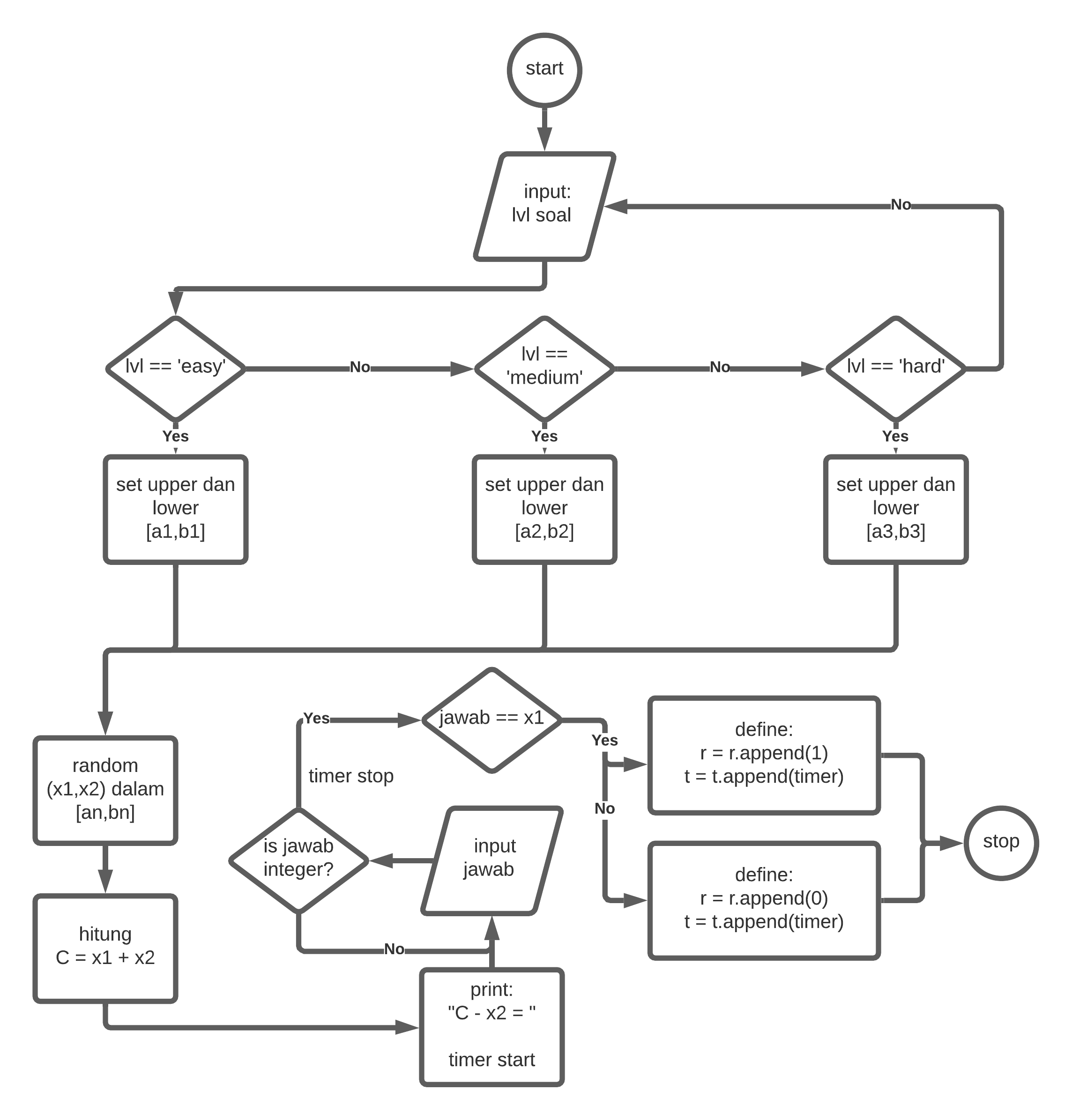
Flowchart Utama

### *Flowchart Sub Algoritma*

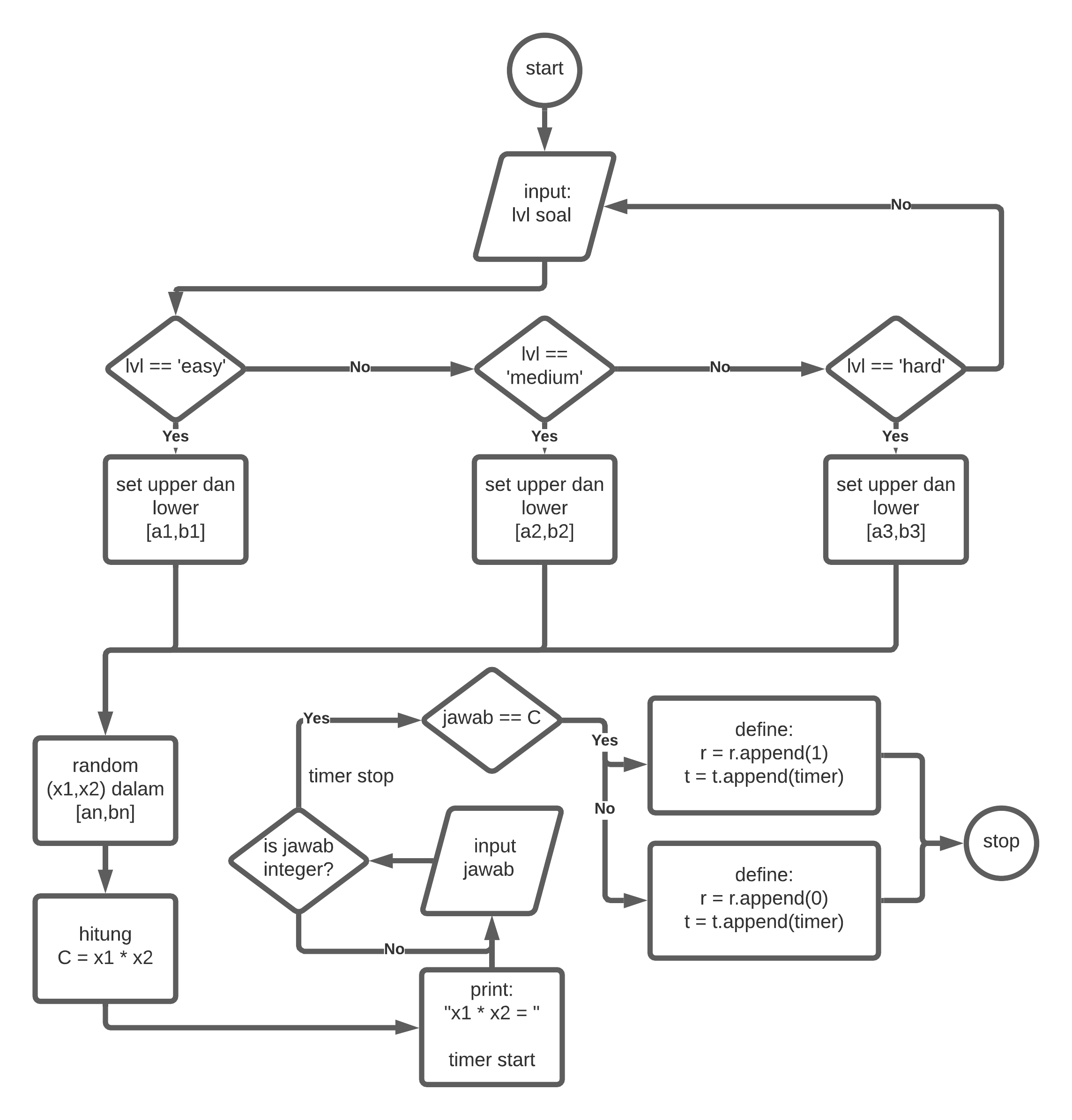
Berikut adalah *flowchart* dari semua sub algoritma yang ada. Khusus pada *flowchart* penjumlahan, pengurangan, perkalian, dan pembagian: **angka** \_**random** **yang dibuat oleh** ***function*** randint() **adalah bilangan bulat**.



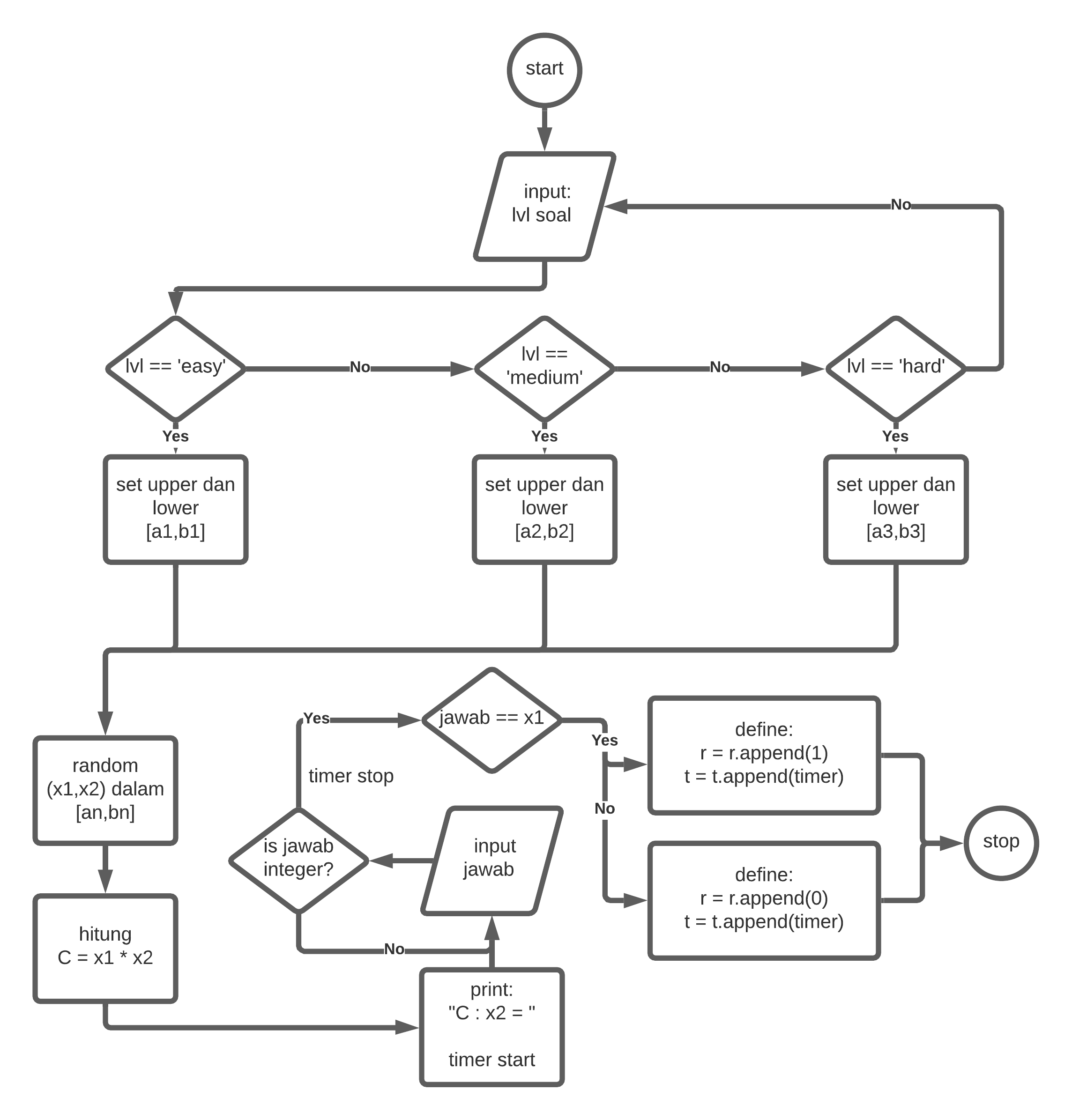
Flowchart Penjumlahan



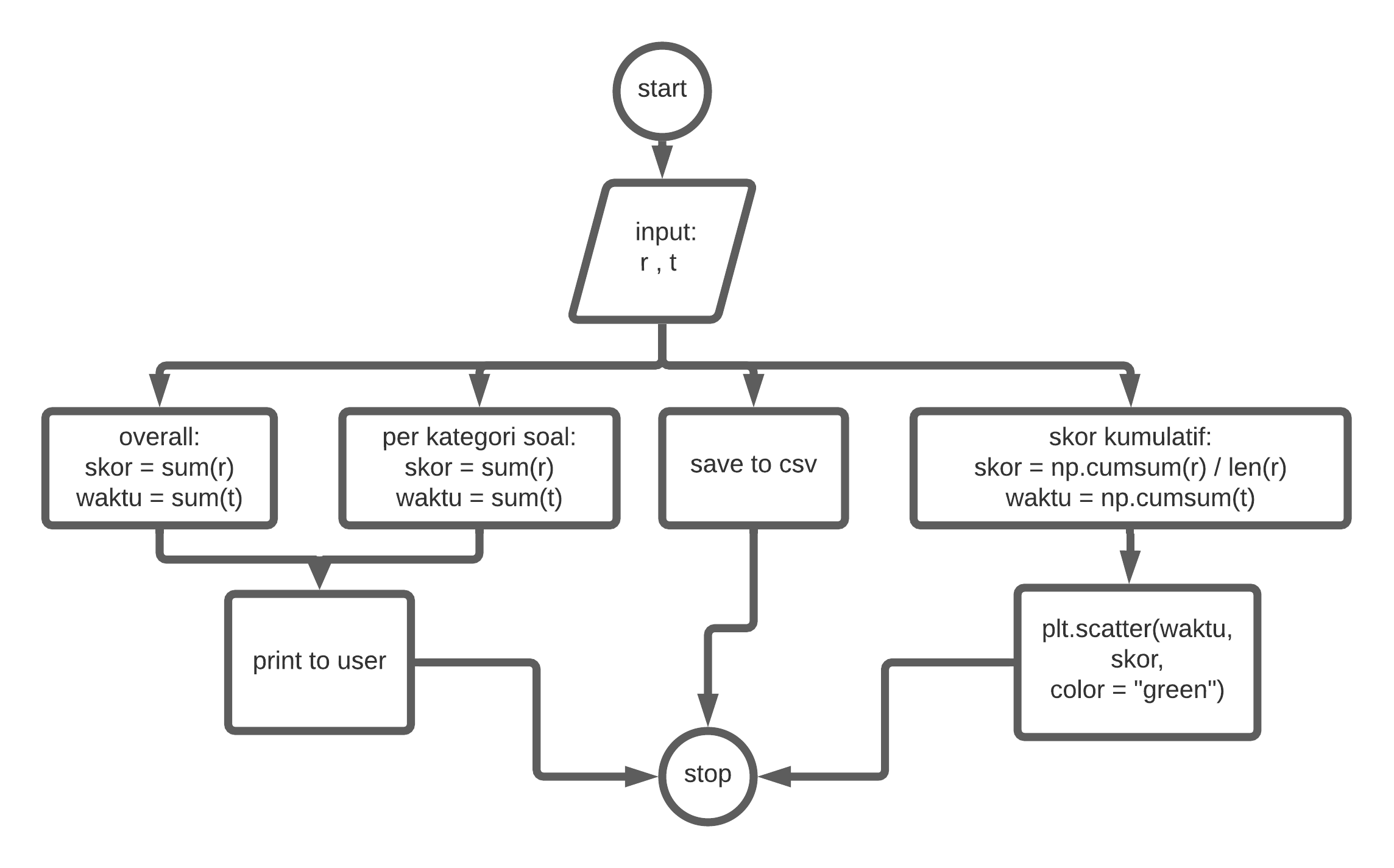
Flowchart Pengurangan



Flowchart Perkalian



Flowchart Pembagian



Flowchart Rekap dan Plot

*Codes* dari program ini disimpan dalam dua *files*, yakni:

1. program.py berisi *codes* dari *flowchart* utama dan semua *flowchart* sub algoritma.
2. main.py merupakan program yang akan dieksekusi oleh *Python3* saat:
   * Anak mengeksekusi program **pertama kali**, ATAU
   * Anak mau **mencoba kembali** setelah menyelesaikan keseluruhan rangkaian soal.

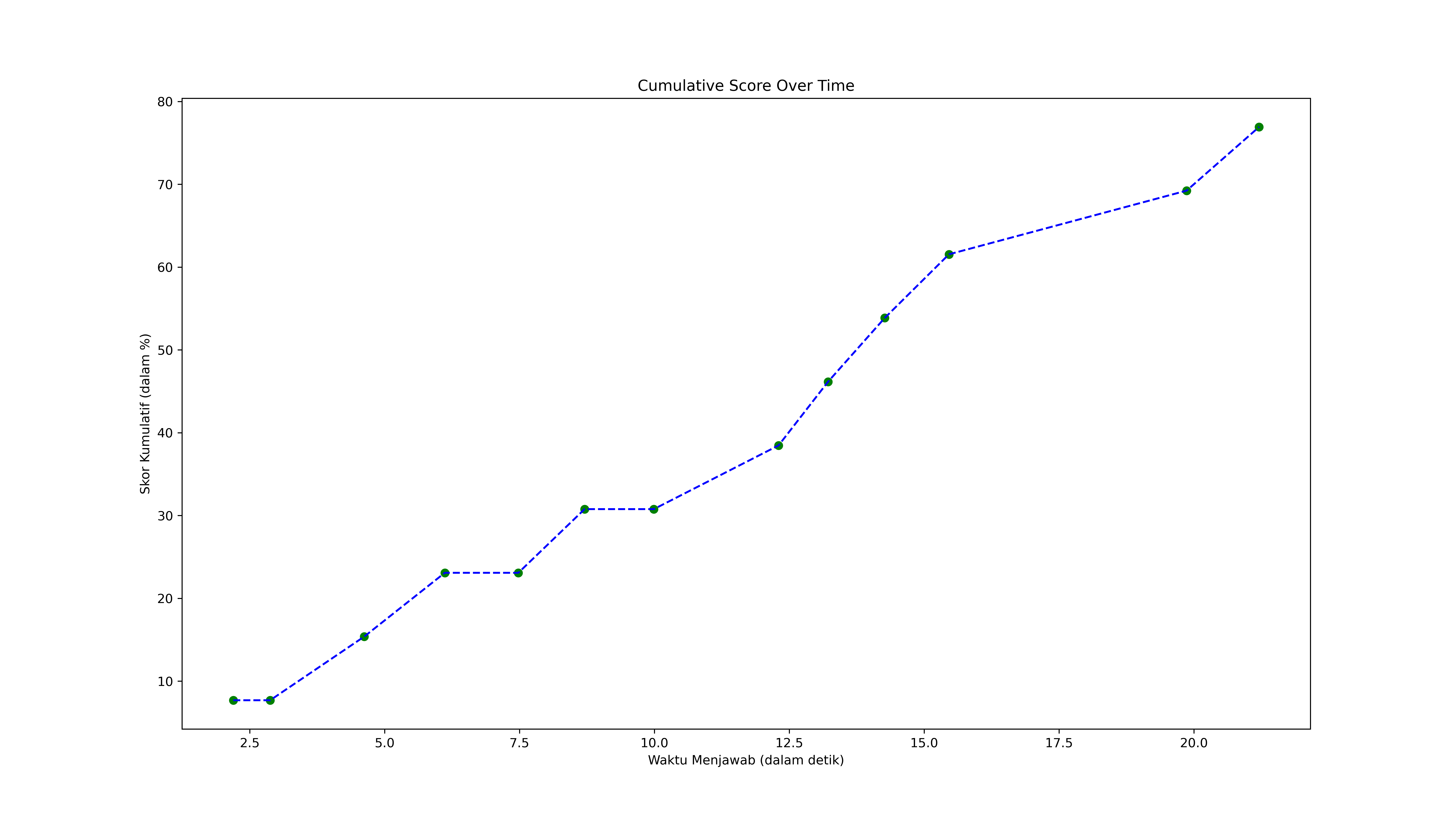
Isi dari main.py adalah sebagai berikut:

# set initial condition  
lagi = "y"  
# menanyakan apakah user mau coba lagi?  
while lagi == "y" or lagi == "Y":  
 exec(open("program.py").read())  
 lagi = input("Coba lagi? (y/n)")  
print("==== Terima kasih ====")

Kedua *files* ini akan dilampirkan dalam laporan ini dan dapat diakses di halaman *Github Repository* berikut: <https://github.com/ikanx101/RBL-SK5003>

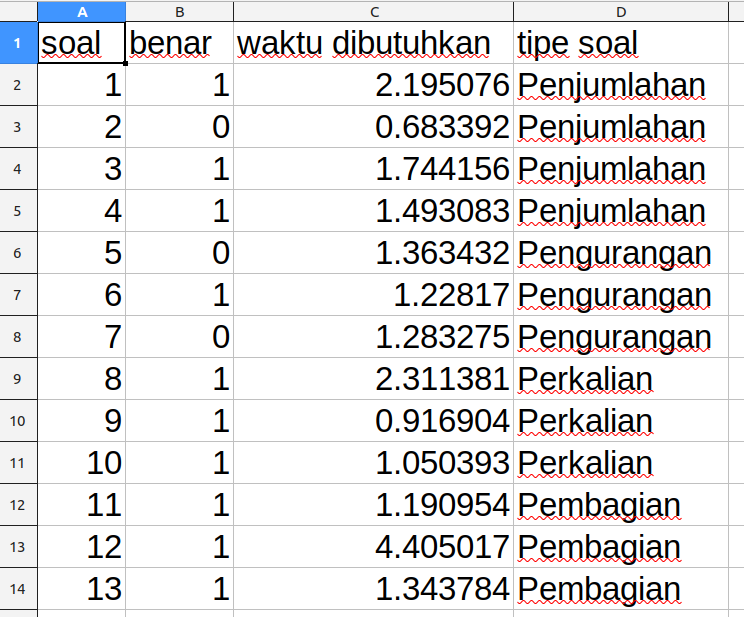
### Contoh *Output* dari Program CLI

Berikut adalah contoh grafik rekap jawaban anak:



Contoh Plot Rekap Jawaban

Berikut adalah contoh file .csv dari jawaban anak:



Contoh csv Data Rekap Jawaban

# REFERENSI

[1] McCrindle, M., Fell, A. (2020) Understanding Generation Alpha. McCrindle Research Pty Ldt.

[2] McGugan, Will (2007) Beginning Game Development with Python dan Pygame from Novice to Professional. Apress.

# LAMPIRAN

## *Codes* **program.py**

"""  
=================================  
RESEARCH-BASED LEARNING  
SK5003 - PEMROGRAMAN DALAM SAINS  
=================================  
  
Kelompok RBL 2  
20921003 - Winda Wijayasari  
20921004 - Mohammad Rizka Fadhli  
  
Program ini bertujuan utk generating random soal  
berhitung untuk anak-anak kelas 2-4 SD.  
"""  
  
# ==============================================================  
# penentuan level berdasarkan angka yang akan digenerate:  
 # penjumlahan dan pengurangan:   
 # easy: 1-30  
 # medium: 30 - 70  
 # hard: 70 - 929  
 # perkalian dan pembagian:  
 # easy: 1-8  
 # medium: 4-12  
 # hard: 8-20  
# ==============================================================  
  
# fungsi untuk clear halaman  
def clear():  
 for i in range(0,10):  
 print("\n\n")  
  
  
# import fungsi yang dibutuhkan  
from random import randint # digunakan untuk melihat waktu  
import datetime # untuk melihat waktu  
import matplotlib.pyplot as plt # untuk melakukan plot  
import numpy as np  
  
  
clear()  
print('Halo,\n\nSelamat datang di program generator soal Matematika di Python.\nUntuk memulai, silakan masukkan level kesulitan yang diinginkan:\n')  
  
# fungsi level  
# ini masih sederhana ya utk penjumlahan terlebih dahulu  
def level():  
 print('Level kesulitan soal:\n1. easy\n2. medium\n3. hard\n\n(Masukkan angka level)\n\n')  
 # digunakan untuk mengecek apakah inputnya berupa integer atau tidak  
 marker = True  
 while marker:  
 try:  
 lvl = int(input("level: "))  
 if lvl > 3 or lvl < 1:  
 marker = True  
 print("Masukkan angka sesuai instruksi")  
 else: break  
 except ValueError:  
 print("Format input salah. Masukkan angka sesuai instruksi.\n\n")  
  
 if lvl == 1:  
 x1 = 1  
 x2 = 30  
 x3 = 1  
 x4 = 8  
 if lvl == 2:  
 x1 = 30  
 x2 = 70  
 x3 = 4  
 x4 = 12  
 if lvl == 3:  
 x1 = 70  
 x2 = 200  
 x3 = 8  
 x4 = 20  
 out = [x1,x2,x3,x4]  
 return(out)  
  
# fungsi untuk menentukan n soal  
def n\_soal():  
 # memastikan bahwa hanya bleh masuk n berupa integer > 0  
 marker = True  
 while marker:  
 try:  
 n = int(input("Banyaknya soal yang mau dikerjakan: "))  
 if n < 0:  
 marker = True  
 print("Masukkan angka bilangan bulat positif.\n\n")  
 else: break  
 except ValueError:  
 print("Masukkan angka sesuai instruksi (bilangan bulat).\n\n")  
 return(n)  
  
# fungsi penjumlahan  
def jumlah(x1,x2) :  
 a = randint(x1,x2)  
 b = randint(x1,x2)  
 c = a + b  
 print(a, ' + ', b, ' = ')  
 now = datetime.datetime.now()  
   
 # memastikan agar hanya integer yang boleh diinput  
 marker = True  
 while marker:  
 try:  
 jawab = int(input("jawab: "))  
 break  
 except ValueError:  
 print("Masukkan hanya angka!\n")  
   
 end = datetime.datetime.now()  
 time\_delta = end-now  
 total\_waktu = time\_delta.total\_seconds()  
 jawab = int(jawab)  
 if c == jawab :  
 cek = 1  
 # print(cek)  
 else :  
 cek = 0  
 # print(cek)  
 return(cek,total\_waktu)  
  
# fungsi pengurangan  
def kurang(x1,x2) :  
 a = randint(x1,x2)  
 b = randint(x1,x2)  
 c = a + b  
 print(c, ' - ', a, ' = ')  
 now = datetime.datetime.now()  
   
 # memastikan agar hanya integer yang boleh diinput  
 marker = True  
 while marker:  
 try:  
 jawab = int(input("jawab: "))  
 break  
 except ValueError:  
 print("Masukkan hanya angka!\n")  
   
 end = datetime.datetime.now()  
 time\_delta = end-now  
 total\_waktu = time\_delta.total\_seconds()  
 jawab = int(jawab)  
 if b == jawab :  
 cek = 1  
 # print(cek)  
 else :  
 cek = 0  
 # print(cek)  
 return(cek,total\_waktu)  
  
# fungsi perkalian  
def kali(x3,x4) :  
 a = randint(x3,x4)  
 b = randint(x3,x4)  
 c = a \* b  
 print(a, ' x ', b, ' = ')  
 now = datetime.datetime.now()  
   
 # memastikan agar hanya integer yang boleh diinput  
 marker = True  
 while marker:  
 try:  
 jawab = int(input("jawab: "))  
 break  
 except ValueError:  
 print("Masukkan hanya angka!\n")  
   
 end = datetime.datetime.now()  
 time\_delta = end-now  
 total\_waktu = time\_delta.total\_seconds()  
 jawab = int(jawab)  
 if c == jawab :  
 cek = 1  
 # print(cek)  
 else :  
 cek = 0  
 # print(cek)  
 return(cek,total\_waktu)  
  
# fungsi pembagian  
def bagi(x3,x4) :  
 a = randint(x3,x4)  
 b = randint(x3,x4)  
 c = a \* b  
 print(c, ' : ', a, ' = ')  
 now = datetime.datetime.now()  
   
 # memastikan agar hanya integer yang boleh diinput  
 marker = True  
 while marker:  
 try:  
 jawab = int(input("jawab: "))  
 break  
 except ValueError:  
 print("Masukkan hanya angka!\n")  
   
 end = datetime.datetime.now()  
 time\_delta = end-now  
 total\_waktu = time\_delta.total\_seconds()  
 jawab = int(jawab)  
 if b == jawab :  
 cek = 1  
 # print(cek)  
 else :  
 cek = 0  
 # print(cek)  
 return(cek,total\_waktu)  
  
  
# =========================================  
# tampilannya mulai dari sini  
  
# penentuan level kesulitan  
lv = level()  
  
# bikin array untuk rekap  
benar = [] # berapa banyak jawaban benar  
waktu = [] # waktu yang dibutuhkan  
tipe\_soal = [] # tipe soal  
  
  
# menentukan seberapa banyak soal yang dikerjakan  
print("\n\nPENJUMLAHAN\n")  
n\_penjumlahan = n\_soal()  
n\_penjumlahan = int(n\_penjumlahan) - 1  
  
# menentukan seberapa banyak soal yang dikerjakan  
print("\n\nPENGURANGAN\n")  
n\_pengurangan = n\_soal()  
n\_pengurangan = int(n\_pengurangan) - 1  
  
# menentukan seberapa banyak soal yang dikerjakan  
print("\n\nPERKALIAN\n")  
n\_perkalian = n\_soal()  
n\_perkalian = int(n\_perkalian) - 1  
  
# menentukan seberapa banyak soal yang dikerjakan  
print("\n\nPEMBAGIAN\n")  
n\_pembagian = n\_soal()  
n\_pembagian = int(n\_pembagian) - 1  
  
clear()  
  
# mulai iterasi untuk penjumlahan  
i = 0  
tes = []  
jumlah\_benar = []  
while i <= n\_penjumlahan :  
 tes = jumlah(lv[0],lv[1])  
 print("\n")  
 benar.append(tes[0])  
 waktu.append(tes[1])  
 jumlah\_benar.append(tes[0])  
 tipe\_soal.append("Penjumlahan")  
 i = i + 1  
  
# mulai iterasi untuk pengurangan  
i = 0  
kurang\_benar = []  
while i<=n\_pengurangan :  
 tes = kurang(lv[0],lv[1])  
 print("\n")  
 benar.append(tes[0])  
 waktu.append(tes[1])  
 kurang\_benar.append(tes[0])  
 tipe\_soal.append("Pengurangan")  
 i = i + 1  
  
# mulai iterasi untuk perkalian  
i = 0  
kali\_benar = []  
while i<=n\_perkalian :  
 tes = kali(lv[2],lv[3])  
 print("\n")  
 benar.append(tes[0])  
 waktu.append(tes[1])  
 kali\_benar.append(tes[0])  
 tipe\_soal.append("Perkalian")  
 i = i + 1  
  
# mulai iterasi untuk pembagian  
i = 0  
bagi\_benar = []  
while i<=n\_pembagian :  
 tes = bagi(lv[2],lv[3])  
 print("\n")  
 benar.append(tes[0])  
 waktu.append(tes[1])  
 bagi\_benar.append(tes[0])  
 tipe\_soal.append("Pembagian")  
 i = i + 1  
  
# output ke user  
print("\n\n")  
print("Kamu berhasil menjawab: ",sum(benar)," soal dari total ",len(benar)," soal")  
print("Waktu yang dibutuhkan: ",round(sum(waktu),4)," detik")  
  
# rekap penjumlahan  
print("\n")  
jumlah\_benar = sum(jumlah\_benar)  
print("Kamu berhasil menjawab ",jumlah\_benar," soal penjumlahan dari ",n\_penjumlahan+1," soal")  
  
# rekap pengurangan  
print("\n")  
kurang\_benar = sum(kurang\_benar)  
print("Kamu berhasil menjawab ",kurang\_benar," soal pengurangan dari ",n\_pengurangan+1," soal")  
  
# rekap perkalian  
print("\n")  
kali\_benar = sum(kali\_benar)  
print("Kamu berhasil menjawab ",kali\_benar," soal perkalian dari ",n\_perkalian+1," soal")  
  
# rekap pembagian  
print("\n")  
bagi\_benar = sum(bagi\_benar)  
print("Kamu berhasil menjawab ",bagi\_benar," soal pembagian dari ",n\_pembagian+1," soal")  
print("\n\nGrafik skor dan rekap berupa csv sudah tersedia di working directory.\n\n")  
  
# bikin rekap berupa file csv  
# memberikan nama file  
f = open("rekap\_jawaban.csv","w+")  
f.write("soal,benar,waktu dibutuhkan,tipe soal\n")  
for j in range(0,len(benar)) :  
 # print (j,". benar = ",benar[j],"; waktu\_dibutuhkan = ",waktu[j],"; tipe soal = ",tipe\_soal[j])  
 f.write(str(j+1)+","+str(benar[j])+","+str(waktu[j])+","+str(tipe\_soal[j])+"\n")  
f.close()  
  
# bikin plot  
benar\_cum = np.cumsum(benar)/len(benar) \* 100  
benar\_cum = np.round(benar\_cum,decimals=2)  
waktu\_cum = np.cumsum(waktu)  
  
  
plt.figure(figsize = (16,9))  
plt.plot(waktu\_cum,benar\_cum,color = "blue",linewidth=1.5,linestyle="--",label='Skor')  
plt.scatter(waktu\_cum,benar\_cum,color = "green")  
plt.xlabel('Waktu Menjawab (dalam detik)')  
plt.ylabel('Skor Kumulatif (dalam %)')  
plt.title("Cumulative Score Over Time")  
#plt.legend()  
#plt.show()  
plt.savefig('rekap.png',dpi = 450)

## *Codes* **main.py**

# ========================================  
# program yang akan dieksekusi oleh user  
# kelompok rbl 2  
# 20921003 Winda Wijayasari  
# 20921004 Mohammad Rizka Fadhli  
# ========================================  
  
# jika user baru pertama kali mengeksekusi  
lagi = "y"  
while lagi == "y" or lagi == "Y":  
 exec(open("program.py").read())  
 # ditanyakan jika user hendak mencoba kembali atau tidak?  
 lagi = input("Coba lagi? (y/n) \n")  
  
print("\n\n")  
print("==== Terima kasih ====")

1. <https://www.stem.org.uk> [↑](#footnote-ref-27)
2. unbk.kemendikbud.com [↑](#footnote-ref-29)
3. Wikipedia [↑](#footnote-ref-31)