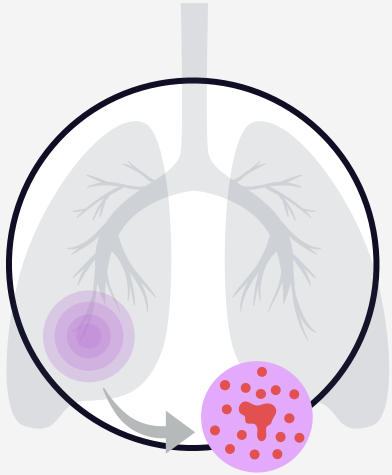
GCPP SYSTEM

(General Cancer Probability Prediction)

#NEVER LATE TO BE AWARE!

GROUP 2
2 ISA 2



THE DEVELOPER

Rafie Ananda Suhermawan (2120010299)

DEVELOPER

Riki Awal Syahputra (2120010136)

LEADER

Alfisyahrin Gusman (2120010016)

DEVELOPER

TABLE OF CONTENTS

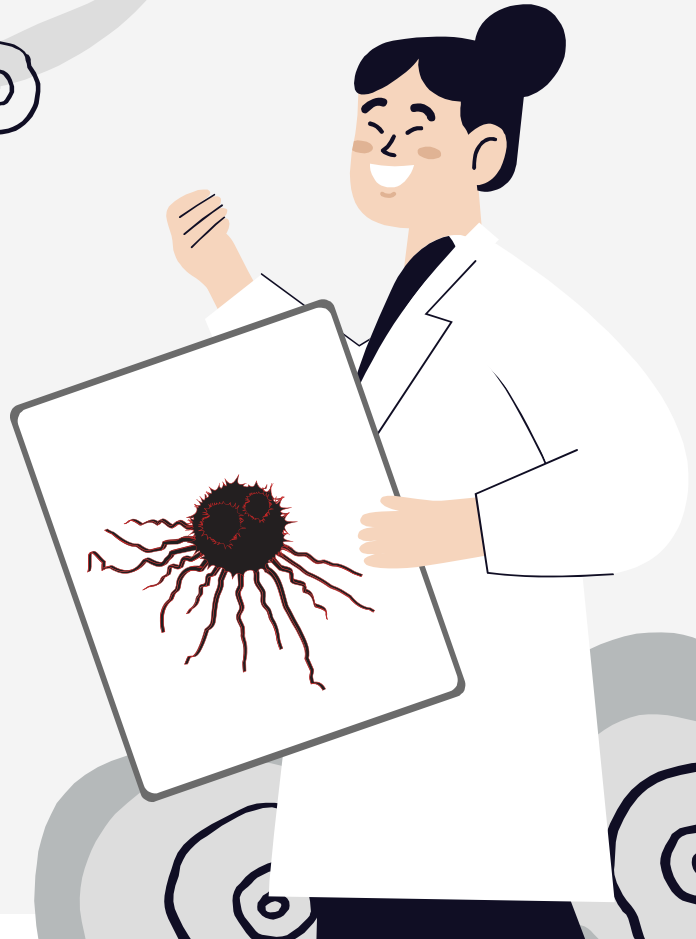
“I had no specific bent toward science until my grandfather died of stomach cancer. I decided that nobody should suffer that much..”

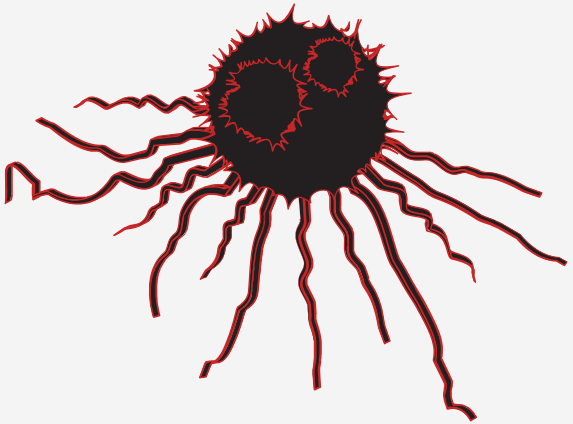
—GERTRUDE B. ELION

01

DEFINITION

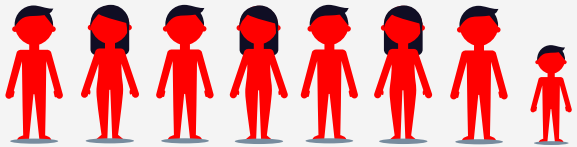
What is GCPP System, how it works and for what?





What is Cancer

A **disease in which some of the body's cells grow uncontrollably and spread to other parts of the body**. Cancer can start almost anywhere in the human body, which is made up of trillions of cells

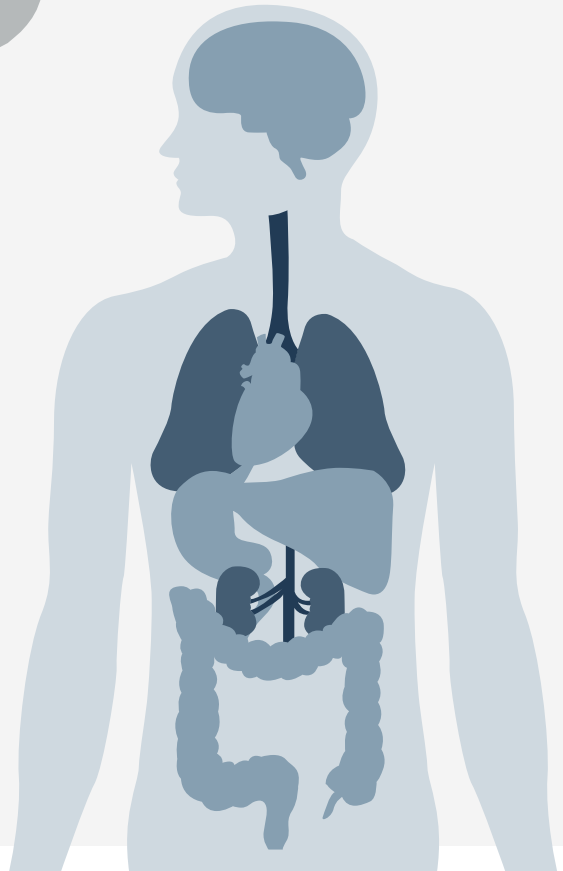


Accounts for 7.4 million deaths worldwide.

What is GCPP System

The system to predict the probability of covering potential cancer risk in general (not specific) and not for the diagnosis, reference data is covered through cancer.net, halodoc.com, and ATSDR (Agency for Toxic Disease Registry)

PARAMETER



- AGE
- GENETIC
- ENVIRONMENT
- LIFESTYLE
- JOB
- DISEASE
- INFECTION
- VACCINATION

Example of GCPP Result

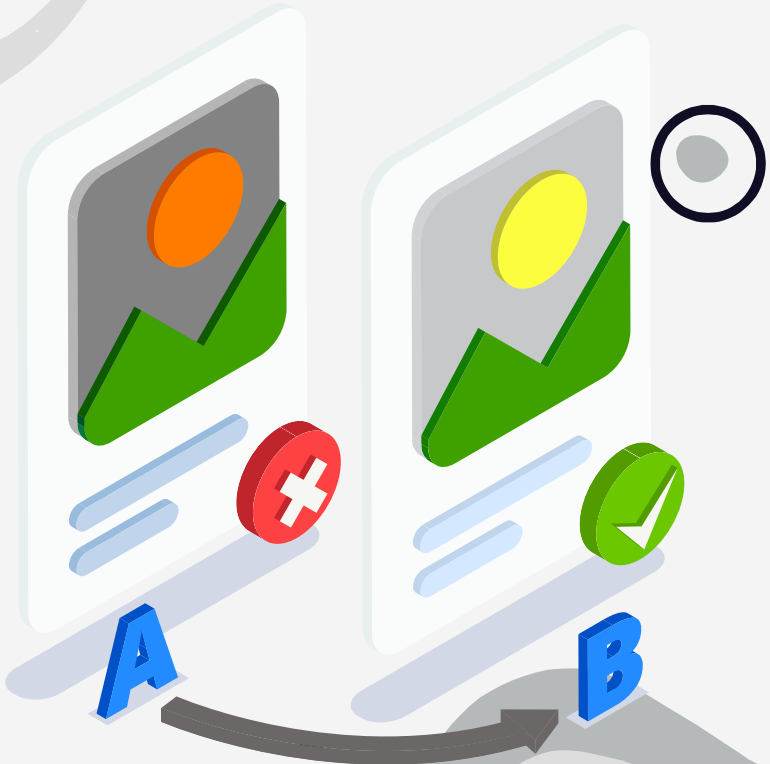
AGE:	19 years
GENDER:	Male
DOMICILE:	South Jakarta
CANCER RISK SCORE:	37.59%

Genetic **Lifestyle** **Infection**

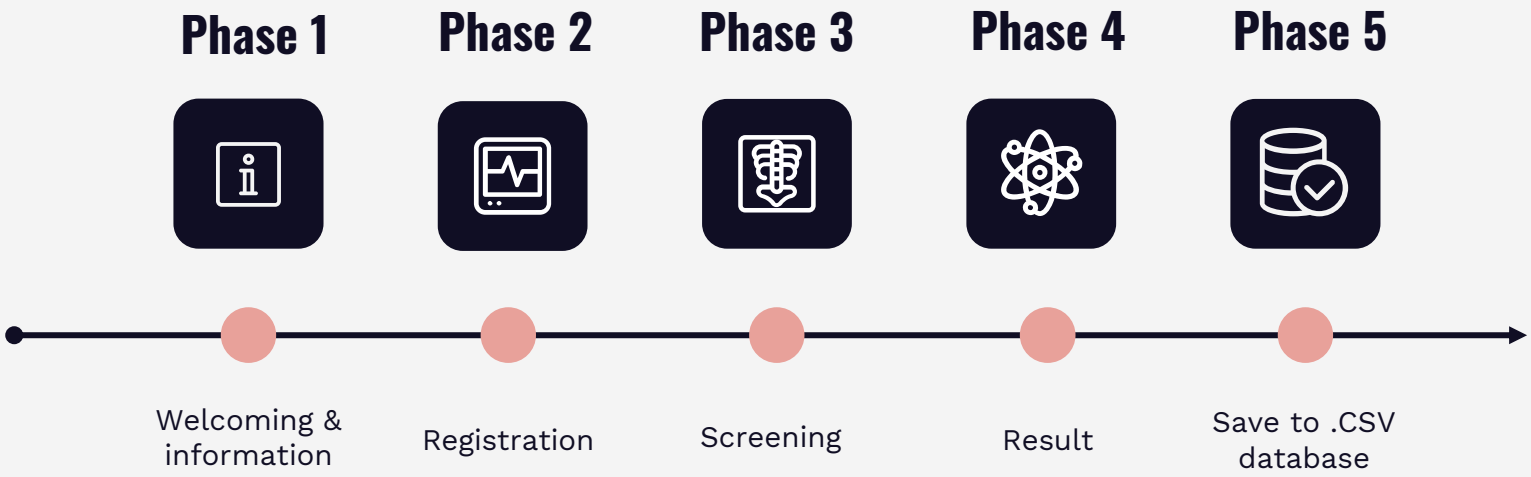
02

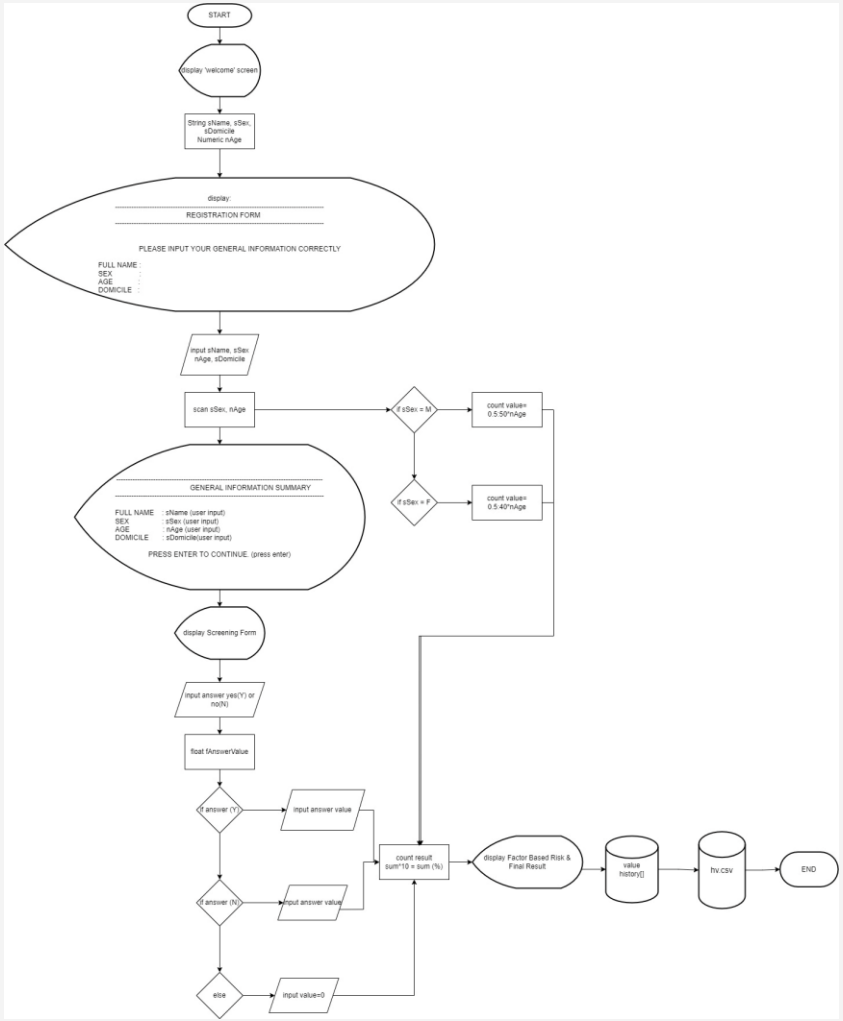
FLOWCHART

How it flows in the program



FUNCTION FLOW





GCPP Flowchart in General

03

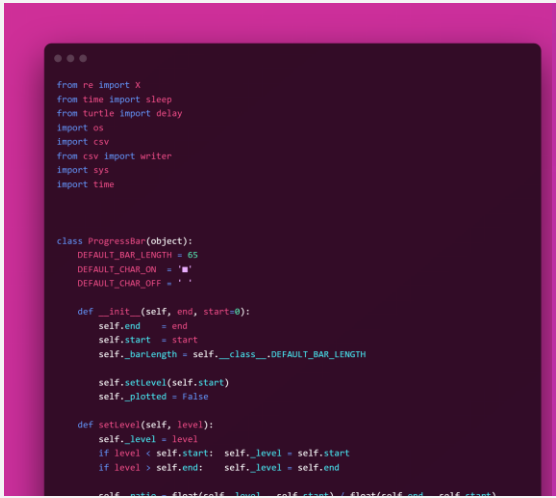
THE CODE AND RESULT

Let’s take a look!

```
        writer_object = writer(f)
        writer_object.writerow(list_data)
        # Close the file object
        f.close()


def start():

    os.system('cls||clear')
    wellcome()
    os.system('cls||clear')
    regist()
    os.system('cls||clear')
    screen()
    os.system('cls||clear')
    res()
    dbs()

    print("\t\t         THIS IS THE END OF THE SYSTEM ")
    dcc = input("\t  SELECT YOUR DECISION [L : RE-START SYSTEM | LL : END SYSTEM] : [")

    if dcc == 'L' or dcc == 'l':
        start()
    if dcc == 'LL' or dcc == 'll':
        exit()

start()
```

WHOLE PROCESS FUNCTION

INPUT LIBRARY

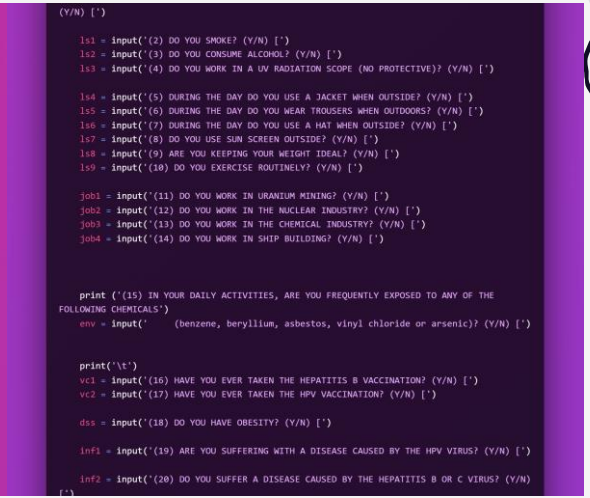


```
from re import X
from time import sleep
from turtle import delay
import os
import csv
from csv import writer
import sys
import time


class ProgressBar(object):
    DEFAULT_BAR_LENGTH = 65
    DEFAULT_CHAR_ON  = '■'
    DEFAULT_CHAR_OFF = ' '

    def __init__(self, end, start=0):
        self.end    = end
        self.start  = start
        self._barLength = self.__class__.DEFAULT_BAR_LENGTH

        self.setLevel(self.start)
        self._plotted = False

    def setLevel(self, level):
        self._level = level
        if level < self.start:  self._level = self.start
        if level > self.end:    self._level = self.end
```

CLASS FUNCTION TO STORE DATA

```
xvc = []
xinf = []

def _nm(self,x):
    self.x = x
    inp.nm.append(x)
def _sex(self,x):
    self.x = x
    inp.sex.append(x)
def _age(self,x):
    self.x = x
    inp.age.append(int(x))
def _dm(self,x):
    self.x = x
    inp.dm.append(x)

def _agec(self,x):
    self.x = x
    inp.agec.append(x)
def _dsc(self,x):
    self.x = x
    inp.dsc.append(x)
def _ls(self,x):
    self.x = x
    inp.ls.append(x)
def _job(self,x):
    self.x = x
    inp.job.append(float(x))
def _env(self,x):
    self.x = x
    inp.env.append(x)

def _vc(self,x):
    self.x = x
    inp.vc.append(x)
```

THE QUESTION AND VARIABLE INPUT

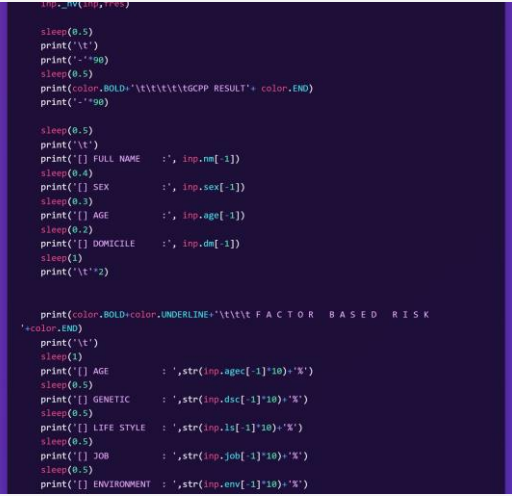


```
(Y/N) [')

    ls1 = input('(2) DO YOU SMOKE? (Y/N) [')
    ls2 = input('(3) DO YOU CONSUME ALCOHOL? (Y/N) [')
    ls3 = input('(4) DO YOU WORK IN A UV RADIATION SCOPE (NO PROTECTIVE)? (Y/N) [')

    ls4 = input('(5) DURING THE DAY DO YOU USE A JACKET WHEN OUTSIDE? (Y/N) [')
    ls5 = input('(6) DURING THE DAY DO YOU WEAR TROUSERS WHEN OUTDOORS? (Y/N) [')
    ls6 = input('(7) DURING THE DAY DO YOU USE A HAT WHEN OUTSIDE? (Y/N) [')
    ls7 = input('(8) DO YOU USE SUN SCREEN OUTSIDE? (Y/N) [')
    ls8 = input('(9) ARE YOU KEEPING YOUR WEIGHT IDEAL? (Y/N) [')
    ls9 = input('(10) DO YOU EXERCISE ROUTINELY? (Y/N) [')

    job1 = input('(11) DO YOU WORK IN URANIUM MINING? (Y/N) [')
    job2 = input('(12) DO YOU WORK IN THE NUCLEAR INDUSTRY? (Y/N) [')
    job3 = input('(13) DO YOU WORK IN THE CHEMICAL INDUSTRY? (Y/N) [')
    job4 = input('(14) DO YOU WORK IN SHIP BUILDING? (Y/N) [')



    print ('(15) IN YOUR DAILY ACTIVITIES, ARE YOU FREQUENTLY EXPOSED TO ANY OF THE
FOLLOWING CHEMICALS')
    env = input('     (benzene, beryllium, asbestos, vinyl chloride or arsenic)? (Y/N) [')


    print('\t')
    vc1 = input('(16) HAVE YOU EVER TAKEN THE HEPATITIS B VACCINATION? (Y/N) [')
    vc2 = input('(17) HAVE YOU EVER TAKEN THE HPV VACCINATION? (Y/N) [')

    dss = input('(18) DO YOU HAVE OBESITY? (Y/N) [')

    inf1 = input('(19) ARE YOU SUFFERING WITH A DISEASE CAUSED BY THE HPV VIRUS? (Y/N) [')

    inf2 = input('(20) DO YOU SUFFER A DISEASE CAUSED BY THE HEPATITIS B OR C VIRUS? (Y/N)
[')
```

SIMPLE DECISION TO ADD VALUE

```
    inf3 = input('(21) ARE YOU SUFFERING WITH A DISEASE CAUSED BY THE EPSTEIN-BARR VIRUS?
(Y/N) [')
    inf4 = input('(22) ARE YOU SUFFERING WITH A DISEASE CAUSED BY H. PYLORI BACTERIA?
(Y/N) [')



    #DSC
    if dsc == 'y' or dsc == 'Y':
        inp._val(inp,0.544782608695652)
        inp._dsc(inp,0.544782608695652)
    else:
        inp._val(inp,0)
        inp._dsc(inp,0)



    #LS
    if ls1 == 'y' or ls1 == 'Y':
        inp._val(inp,0.424782608695651)
        inp.xls.append(0.424782608695651)
    else:
        inp._val(inp,0)
        inp.xls.append(0)

    if ls2 == 'y' or ls2 == 'Y':
        inp._val(inp,0.414782608695651)
        inp.xls.append(0.414782608695651)
    else:
        inp._val(inp,0)
        inp.xls.append(0)

    if ls3 == 'y' or ls3 == 'Y':
        inp._val(inp,0.404782608695651)
```

SHOWING THE RESULT OF CALCULATION



```
    sleep(0.5)
    print('\t')
    print('-'*90)
    sleep(0.5)
    print(color.BOLD+'\t\t\t\t\tGCPP RESULT'+ color.END)
    print('-'*90)

    sleep(0.5)
    print('\t')
    print('[] FULL NAME    :', inp.nm[-1])
    sleep(0.4)
    print('[] SEX          :', inp.sex[-1])
    sleep(0.3)
    print('[] AGE          :', inp.age[-1])
    sleep(0.2)
    print('[] DOMICILE     :', inp.dm[-1])
    sleep(1)
    print('\t'*2)


    print(color.BOLD+color.UNDERLINE+'\t\t\t F A C T O R   B A S E D   R I S K
'+color.END)
    print('\t')
    sleep(1)
    print('[] AGE          : ',str(inp.agec[-1]*10)+'%')
    sleep(0.5)
    print('[] GENETIC      : ',str(inp.dsc[-1]*10)+'%')
    sleep(0.5)
    print('[] LIFE STYLE   : ',str(inp.ls[-1]*10)+'%')
    sleep(0.5)
    print('[] JOB          : ',str(inp.job[-1]*10)+'%')
    sleep(0.5)
    print('[] ENVIRONMENT  : ',str(inp.env[-1]*10)+'%')
```

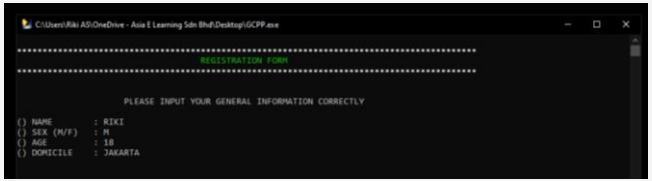
STORE DATA TO .CSV DATABASE

```
    sleep(0.5)
    print('[] VACCINATION  : ',str(inp.vc[-1]*10)+'%')
    sleep(0.5)
    print('[] DISEASE      : ',str(inp.dss[-1]*10)+'%')
    sleep(0.5)
    print('[] INFECTION    : ',str(inp.inf[-1]*10)+'%')
    sleep(1.5)
    print('\t')



    print(color.YELLOW+color.BOLD+color.UNDERLINE+'\t\t\t C A N C E R   R I S K   S C O R
E '+color.END)
    sleep(2)
    print('\t')
    print('\t\t\t\t',fres)
    print('\t')
    print('.'*90)

def dbs():

    # The data assigned to the list
    list_data=[inp.nm[-1], inp.sex[-1], inp.age[-1], inp.dm[-1], inp.agec[-1],
inp.dsc[-1], inp.ls[-1], inp.job[-1], inp.env[-1], inp.vc[-1], inp.dss[-1], inp.inf[-1],
inp.agec[-1]*10, inp.dsc[-1]*10, inp.ls[-1]*10, inp.job[-1]*10, inp.env[-1]*10,
inp.vc[-1]*10, inp.dss[-1]*10, inp.inf[-1]*10, inp.hv[-1]]
    header = ['Name', 'Branch', 'Year', 'CGPA']

    f = open('HV.csv', 'a')
    # xy = csv.DictWriter(f, fieldnames = header)
    # xy.writeheader()

    with f:
        f = open('HV.csv', 'a', newline='')
```

THE RESULT

[1]

[2]



[3]

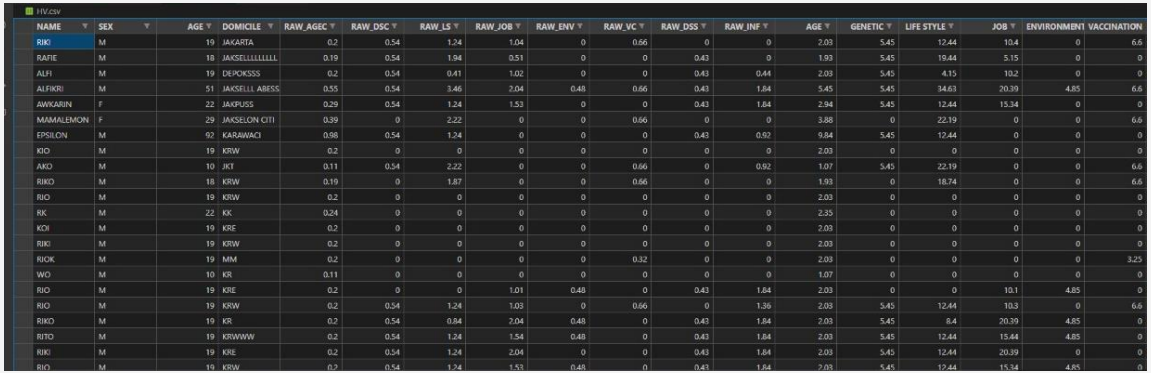
[4]

[5]

[6]

[7]

HV.CSV
DATABASE
(Historical Value)



HV.csv

NAME	SEX	AGE	DOMICILE	RAW_AGEC	RAW_DSC	RAW_LS	RAW_JOB	RAW_ENV	RAW_VC	RAW_DSS	RAW_INF	AGE	GENETIC	LIFE STYLE	JOB	ENVIRONMENT	VACCINATION
RIKI	M	19	JAKARTA	0.2	0.54	1.24	1.04	0	0.66	0	0	2.03	5.45	12.44	10.4	0	6.6
RAFIE	M	18	JAKSELLLLLLLL	0.19	0.54	1.94	0.51	0	0	0.43	0	1.93	5.45	19.44	5.15	0	0
ALFI	M	19	DEPOKSSS	0.2	0.54	0.41	1.02	0	0	0.43	0.44	2.03	5.45	4.15	10.2	0	0
ALFIKRI	M	51	JAKSELLL ABESS	0.55	0.54	3.46	2.04	0.48	0.66	0.43	1.84	5.45	5.45	34.63	20.39	4.85	6.6
AWKARIN	F	22	JAKPUSS	0.29	0.54	1.24	1.53	0	0	0.43	1.84	2.94	5.45	12.44	15.34	0	0
MAMALEMON	F	29	JAKSELON CITI	0.39	0	2.22	0	0	0.66	0	0	3.88	0	22.19	0	0	6.6
EPSILON	M	92	KARAWACI	0.98	0.54	1.24	0	0	0	0.43	0.92	9.84	5.45	12.44	0	0	0
KIO	M	19	KRW	0.2	0	0	0	0	0	0	0	2.03	0	0	0	0	0
AKO	M	10	JKT	0.11	0.54	2.22	0	0	0.66	0	0.92	1.07	5.45	22.19	0	0	6.6
RIKO	M	18	KRW	0.19	0	1.87	0	0	0.66	0	0	1.93	0	18.74	0	0	6.6
RIO	M	19	KRW	0.2	0	0	0	0	0	0	0	2.03	0	0	0	0	0
RK	M	22	KK	0.24	0	0	0	0	0	0	0	2.35	0	0	0	0	0
KOI	M	19	KRE	0.2	0	0	0	0	0	0	0	2.03	0	0	0	0	0
RIKI	M	19	KRW	0.2	0	0	0	0	0	0	0	2.03	0	0	0	0	0
RIOK	M	19	MM	0.2	0	0	0	0	0.32	0	0	2.03	0	0	0	0	3.25
WO	M	10	KR	0.11	0	0	0	0	0	0	0	1.07	0	0	0	0	0
RIO	M	19	KRE	0.2	0	0	1.01	0.48	0	0.43	1.84	2.03	0	0	10.1	4.85	0
RIO	M	19	KRW	0.2	0.54	1.24	1.03	0	0.66	0	1.36	2.03	5.45	12.44	10.3	0	6.6
RIKO	M	19	KR	0.2	0.54	0.84	2.04	0.48	0	0.43	1.84	2.03	5.45	8.4	20.39	4.85	0
RITO	M	19	KRWWW	0.2	0.54	1.24	1.54	0.48	0	0.43	1.84	2.03	5.45	12.44	15.44	4.85	0
RIKI	M	19	KRE	0.2	0.54	1.24	2.04	0	0	0.43	1.84	2.03	5.45	12.44	20.39	0	0
RIO	M	19	KRW	0.2	0.54	1.24	1.53	0.48	0	0.43	1.84	2.03	5.45	12.44	15.34	4.85	0

04

CONFIGURATION

I recommend this!

System Requirement Recomendation

Hardware : Intel I7 gen 11, RAM 16 GB HDD 100 GB or 1 TB, Video graphicNVidia GeForce RTX 3080

VGA Resolution : 7680x4320

Operating System : Windows 11

Software : - Command Prompt

- Microsoft Word 2021
- Microsoft Excel 2021

THANKS!