

Aplikasi Pengukuran Berat dan Tinggi Badan Anak Balita Menggunakan Metode *Radbms* Berbasis *Python*

Juli Sardi¹, Hamdani², Vito Basjika Pramuja³

^{1,2,3}Universitas Negeri Padang

Jl. Prof Dr. Hamka Air Tawar, Padang

JuliSardi@ft.unp.ac.id²

Abstract — This research is making an application for measuring weight and height of toddlers using the python-based realtime active database management system (RADBMS) method. The goal of this final research is to design and build a weight and height measurement application for children under five that can store its database safely using real-time active database management methods. The design and manufacture of this application for measuring weight and height for toddlers is made with python as a programming language for making the application and MySQL as the database. By using the realtime active database management system method, user applications can use stored databases for reuse display toddler growth charts. And the database that is stored in softcopy in the form of a database can be printed as a hardcopy by becoming a file with the extension .xlsx and then printed afterwards. After testing, the application can be used properly starting from the security of the login interface, the biodata that is stored when inputting data on the data input interface, and when making measurements on the measurement interface.

Keywords — Realtime Active Database Management System, Python, MySQL, Database, Softcopy, Hardcopy, Interface, Biodata, Login, Input Data, dan Measurement

Abstrak — Penelitian ini merupakan pembuatan aplikasi pengukuran berat dan tinggi badan anak balita menggunakan metode *realtime active database management system (RADBMS)* berbasis *python*. Yang mana tujuan dari penelitian ini adalah merancang dan membuat aplikasi pengukuran berat dan tinggi bada anak balita yang dapat menyimpan *database*-nya dengan aman menggunakan metode *realtime active dattabase management system*. Perancangan dan pembuatan aplikasi pengukuran berat dan tinggi balita ini dibuat dengan *python* sebagai bahasa pemrograman untuk membuat aplikasi-nya dan MySQL sebagai *database*-nya. Dengan menggunakan metode *realtime active database management system*, pengguna aplikasi ini dapat menggunakan *database* yang tersimpan untuk digunakan kembali menampilkan grafik pertumbuhan balita. Dan *database* yang tersimpan dalam bentuk *softcopy* dalam bentuk *database* bisa dicetak menjadi *hardcopy* dengan mengkonversi menjadi file berekstensi .xlsx dan diprint setelahnya. Setelah dilakukan pengujian aplikasi yang dapat digunakan dengan baik mulai dari keamanan *login interfacenya*, *biodata* yang disimpan saat melakukan *input data* pada *input data interface*, dan saat melakukan pengukuran pada *measurement interface*.

Kata kunci — Realtime Active Database Management System, Python, MySQL, Basis data, Softcopy, Hardcopy, Antarmuka, Biodata, Login, Input Data, dan Pengukuran.

I. PENDAHULUAN

Dewasa ini teknologi yang dipakai pada instansi di daerah pinggiran kota hingga ke desa masih terbilang konvensional. Seperti pengukuran tinggi dan berat balita di Posyandu. Pengukuran tinggi yang dilakukan menggunakan penggaris atau sejenis. Sedangkan untuk pengukuran berat dilakukan dengan timbangan. Berdasarkan kedua pengukuran tadi, sering terjadi *human error* sehingga hasil sebenarnya tidak sesuai karena faktor seperti salah pembacaan oleh mata pengukur. Selain itu, untuk rekam data yang digunakan adalah berupa tulisan tangan ke buku rekam medis. Hal ini membuat hasil rekam data tersebut tidak tersimpan dengan baik dan belum berbentuk *database* sehingga tidak bisa dimanfaatkan untuk proyeksi *trend* grafik kesehatan Anak Balita pada skala yang lebih besar.

Agar lebih efisien, pengukuran tinggi dan berat balita dilakukan dengan alat ukur yang mempunyai presisi yang lebih tinggi. Nantinya alat pengukuran alat ini akan dihasilkan dengan *output* yang jelas, yaitu komputer. Karena pengaplikasian alat ini berbasis komputer. Selain itu, *database* dari hasil pengukuran tadi disimpan di komputer. Sebelumnya, penelitian ini pernah dilakukan dan berhasil dengan baik. Dalam pembuatannya, alat sebelumnya hanya menggunakan PC *desktop* sebagai output hasil ukur dan sumber tegangan bagi alat tersebut untuk hidup [1].

Pada penelitian ini, akan dirancang aplikasi *database* untuk sistem pengukuran dan berat dan tinggi balita pada Posyandu. Penulis akan menggunakan metode atau teknik *Realtime Active Database Management System (RADBS)* [2-4]. *Active Database Management System (ADBMS)* sendiri adalah suatu sistem *database* yang tidak

sekedar menyimpan data tetapi bisa juga melakukan aksi tertentu karena ada suatu *event* dengan memasukkan elemen dinamis serta mempunyai kemampuan memantau *event* untuk mendeteksi ketika data tertentu dimasukkan, dihapus, diubah, atau dipilih kemudian secara otomatis mengeksekusi suatu aksi sebagai respon dari *event* yang terjadi dan kondisi tertentu terpenuhi. Sedangkan *Realtime* yang maksudnya adalah ADBMS tadi bisa dilakukan aksi atau manipulasi tepat pada saat itu juga dan terus berjalan tanpa dipengaruhi waktu.

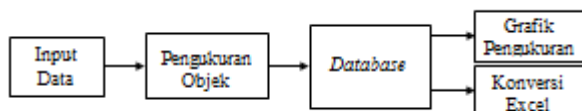
Nantinya, setelah *database* hasil pengukuran tinggi dan berat badan balita disimpan, hasil pengukuran dari pengukuran pertama dan kedua dapat dibandingkan hingga ke bentuk grafik. Biasanya, *database* mendapatkan *input* dari operator manusia, maksudnya adalah dari *input* komputer yang diberikan oleh manusia selaku operator. Seperti penelitian sebelumnya mengenai ADBMS dengan *monitoring* operator [5-8]. Maka dari, penelitian yang pernah dilakukan sebelumnya, penelitian ini dikerjakan dengan *input* pembacaan dari sensor Ultrasonik dan sensor *Load Cell* yang akan dibaca dan diinformasikan kepada Arduino, yang nantinya akan masuk langsung ke *database*.

Program yang digunakan untuk merancang aplikasinya adalah bahasa pemrograman *Python*. Bahasa pemrograman *Python* digunakan karena bahasa pemrograman ini sangat baik dan dianjurkan untuk tujuan analisis data. Bahasa pemrograman *Python* juga lengkap *library*-nya, sehingga bisa digunakan baik untuk membuat *Graphical User Interface* (GUI), menghubungkan ke Arduino, hingga akses ke *database*. Jadi, dalam penelitian ini hanya akan memfokuskan menggunakan bahasa pemrograman *Python* untuk semua pengerjaan. Pada penelitian kali ini, akan difokuskan pekerjaan pembuatan hanya pada aplikasinya [9-10]. Karena, *hardware* yang digunakan adalah *hardware* yang sudah ada pada penelitian sebelumnya. Jadi, pada penelitian ini akan dikembangkan pengaplikasian menjadi lebih efisien hasilnya.

II. METODE

Pada penelitian ini, metode yang digunakan adalah perancangan dan pembuatan aplikasi. Aplikasi ini dirancang untuk digunakan dengan alat pengukur dan berat dan tinggi balita yang telah dijelaskan.

A. Blok Diagram



Gambar 1. Blok Diagram

Blok-Blok diagram perancangan dapat dijelaskan sebagai berikut:

1. *Input Data*

Input Data berfungsi sebagai syarat pertama untuk melakukan pengukuran. Pengukuran tidak dapat dilakukan sebelum input data dilakukan. Nantinya, data

yang telah di-inputkan akan menjadi kata kunci untuk mencari *database* yang diperlukan.

2. Pengukuran Objek

Pengukuran Objek dimana objeknya adalah balita yang akan diukur. Hasil ukur dari pengukuran objek inilah yang akan menjadi *database* berdasarkan dari data yang di-inputkan.

3. *Database*

Database ini berupa biodata yang di-inputkan dan hasil pengukuran dari balita yang diukur. Nantinya, *database* ini dapat difungsikan menjadi grafik pengukuran dan pertumbuhan balita. Selain itu, *database* ini juga dapat dikonversikan menjadi *file excel* untuk dicetak.

4. Grafik Pengukuran

Grafik Pengukuran ini merupakan hasil ukur tinggi dan berat yang dilakukan kepada balita diambil dari *database*. *Database* ini bersifat aktif yang artinya bisa terus dimanfaatkan dan digunakan dimana salah satunya adalah menunjukkan pertumbuhan balita dalam bentuk grafik.

5. Konversi *Excel*

Konversi *Excel* ini berfungsi untuk memanfaatkan *database* biodata beserta hasil ukur dicetak nantinya. *Database* yang ada pada laptop dikonversikan menjadi bentuk *file* berupa format *.xlsx* atau biasa sering disebut *excel*. Lalu setelahnya bisa dicetak bila diperlukan *hardcopy*-nya.

B. Prinsip Kerja Alat

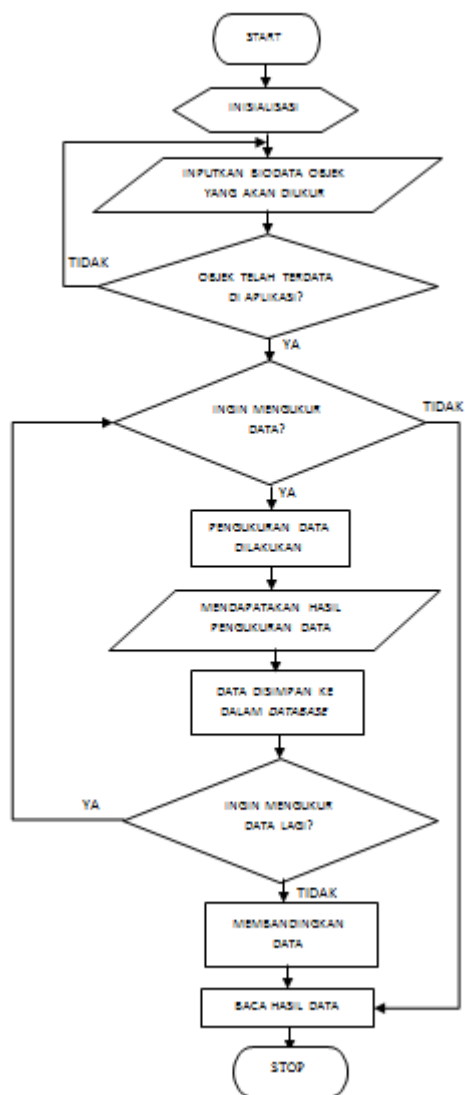
Aplikasi ini dibuat untuk memenuhi kebutuhan pencatatan data hasil pengukuran berat dan tinggi badan balita yang ada di Posyandu. Dengan dibantu *hardware* yang telah dibuat sebelumnya, aplikasi ini dibuat secara *user-friendly* untuk mempermudah orang-orang di Posyandu untuk melakukan pekerjaannya tanpa sulit menggunakannya.

Adapun prinsip kerja alat ini didasarkan pada pengukuran yang dilakukan sensor *Load Cell* untuk mengukur berat badan balita dan sensor *Ultrasonic* untuk mengukur tinggi badan balita. Setelah diukur, hasil pengukuran sementara akan dikirim ke Mikrokontroler Arduino yang nantinya akan disimpan secara permanen pada *database* di penyimpanan memori yang ada pada PC (*Personal Computer*). Setelah data tersimpan, pengguna bisa melakukan aksi lain sesuai pilihannya seperti; mengukur lagi pengukuran yang dilakukan, dan menyelesaikan pengukuran pertama. Pengguna aplikasi ini dapat membuat dan mengubah biodata objek pada aplikasi ini. Pengguna aplikasi ini juga dapat membandingkan data langsung setelah terukur sesuai dengan keinginan yang ingin dilakukan. Aksi *database* yang aktif dan bisa terus-menerus digunakan inilah yang disebut *Realtime Active Database Management System*. Nantinya *database* dapat dimanfaatkan untuk melihat pertumbuhan balita berdasarkan hasil ukur setiap periode waktu yang telah

ditentukan dan bisa pula mengkonversikan *database* menjadi *file* berupa format *.xlsx* yang nantinya bisa dicetak menjadi *hardcopy* bila diperlukan.

C. Flowchart Sistem

Secara sederhana, flowchart cara kerja sistem Aplikasi Pengukuran Berat dan Tinggi atau bisa disebut APBT dapat dilihat pada gambar 2.



Gambar 2. Flowchart Sistem

Aplikasi pengukuran yang akan dibuat didasarkan pada aplikasi yang telah ada sebelumnya, dengan tambahan pengembangan untuk membandingkan data. Aplikasi ini memberikan pilihan bagi pengguna nya untuk melakukan pengukuran data baru, hanya membaca data yang telah tersimpan di database atau membandingkan hasil pengukuran data yang telah dilakukan beberapa kali.

D. Alat Dan Bahan

Adapun alat dan bahan yang digunakan dalam pembuatan aplikasi pengukuran berat dan tinggi badan balita adalah sebagai berikut:

1. Hardware

Hardware yang digunakan dalam pembuatan aplikasi ini adalah penelitian yang telah dibuat pada penelitian sebelumnya [2]. Karena, dalam penelitian ini hanya difokuskan pada pembuatan aplikasinya saja, tidak ikut serta membuat *hardware* nya.

2. Software

Software yang digunakan hanyalah bahasa pemrograman *Python* untuk pembuatan aplikasinya. Dimana bahasa pemrograman *Python* digunakan untuk membuat *Graphical User Interface* (GUI), akses ke *database*, akses ke sistem penginformasian dari *Arduino*, hingga penampilan *output* perbandingan data berupa grafik.

E. Perancangan Umum Sistem

Agar aplikasi ini dapat digunakan dengan baik saat digunakan, aplikasi ini harus dirancang dengan terstruktur. Tentunya dengan perancangan yang baik kesalahan saat penggunaan seperti *error* tidak akan timbul. Adapun perancangan umum sistem dari aplikasi ini adalah sebagai berikut:

1. Login Interface

Login Interface berfungsi sebagai keamanan aplikasi sebelum menggunakan aplikasi ini. Ini dibuat agar tidak semua dan sembarang orang bisa menggunakan aplikasi, karena dalam aplikasi ini berisi *database* penting yang perlu dijaga. Ini juga merupakan salah satu keunggulan aplikasi ini.

2. Input Data Interface

Input Data Interface berfungsi untuk pengisian data / biodata dari balita yang akan diukur. Dalam *interface* ini akan dibuat menu berupa kolom-kolom untuk pengisian data / biodata yang akan diukur. Akan disediakan pula tombol-tombol yang berfungsi untuk membuat data baru, menghapus data, dan memperbarui data yang akan diubah. Pada *interface* akan ditampilkan tabel yang berisi balita yang sudah tersimpan data / biodatanya.

3. Measurement Interface

Measurement Interface atau antarmuka pengukuran adalah *interface* yang berfungsi untuk mengukur balita dan menyimpannya. Akan disediakan tombol-tombol yang berfungsi untuk menyimpan, menghapus dan mengedit hasil ukurnya. Ada pula tombol yang berfungsi untuk melihat grafik pertumbuhan balita berdasarkan hasil ukur yang telah didapat. Juga ada tombol untuk mengkonversikan *database* menjadi *file* dengan format *.xlsx*, ini diperlukan jika ingin mencetak database menjadi *hardcopy* jika diperlukan nantinya.

III. HASIL DAN PEMBAHASAN

Pada bagian ini, akan dilakukan pengujian untuk mendapatkan hasilnya dan untuk dibahas, apakah aplikasinya berjalan dan berfungsi dengan baik dan semestinya atau tidak. Pengujian dilakukan menyangkut beberapa hal sebagai berikut :

A. Pengujian APBT (Aplikasi Pengukur Berat dan Tinggi)

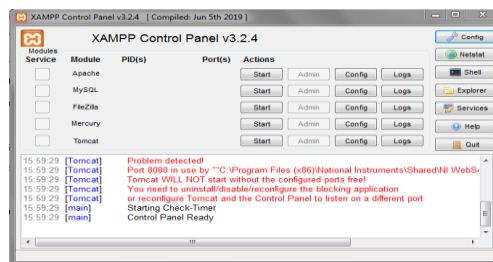
Dalam melakukan pengukuran balita digunakan dua jenis pengujian menggunakan alat yang berbeda, yaitu dengan alat ukur APBT dan alat ukur konvensional. Saat melakukan pengujian, alat ukur APBT digunakan pada pengukuran pertama karena nantinya akan disimpan biodata balita berupa *database* ke laptop. Pengukuran kedua dilakukan agar membandingkan hasil ukur menggunakan alat ukur APBT dan hasil ukur menggunakan alat ukur konvensional yang nilainya adalah nilai pasti atau nilai fakta.

Adapun dalam pengujian alat ukur APBT ini penulis langsung melakukan pengukuran kepada beberapa balita agar dapat memastikan alat ukur ini berfungsi dengan baik atau tidak. Pengujian langsung kepada balita ini juga berfungsi untuk mengimplementasikan langsung tujuan dan manfaat alat ukur APBT ini dibuat.

Cara kerja pengukuran menggunakan alat ukur pada APBT adalah dengan meletakkan objek yang akan diukur diantara pembatas yang telah disediakan. Dimana sensor jarak ultrasonik akan mengukur panjang objek berdasarkan jarak antar pembatas. Karena objek yang akan diukur nantinya adalah balita, maka untuk mengukur balita tersebut balita perlu berbaring diatas alat APBT agar pengukuran tinggi badan akan dilakukan secara horizontal. Maka, nilai panjang yang terukur adalah nilai tinggi balita tersebut. Sedangkan untuk beratnya akan terukur secara otomatis karena *load cell* yang tertanam di dalam alat ukur APBT akan mengukur berat objek yang diletakkan di atasnya.

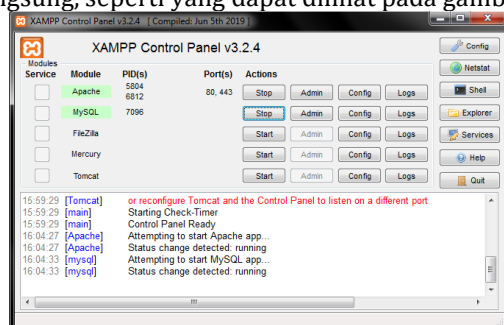
Dalam pengujian alat ukur APBT ini terdapat langkah-langkah cara penggunaannya. Adapun cara langkah-langkah cara menggunakan alat ukur APBT ini adalah sebagai berikut.

1. Pastikan alat ukur APBT seimbang dengan permukaan datar dan tidak terbebani objek apapun sebelum diukur.
2. Hubungkan *serial port* yang ada pada APBT ke laptop / PC.
3. Buka aplikasi *database* pada laptop. Penulis menggunakan aplikasi XAMPP. Klik tombol *Start* pada baris *Apache* dan *Start* pada baris *MySQL* seperti tampak pada gambar 3.



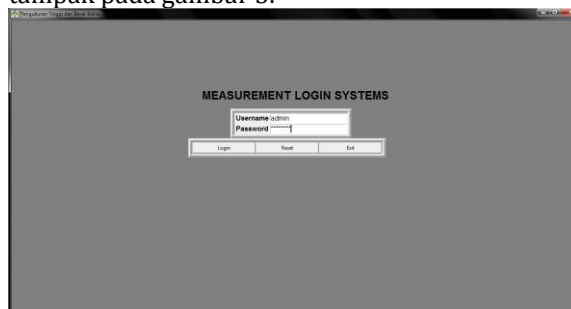
Gambar 3. Control panel XAMPP sebelum terhubung ke *database*

Jika sudah, *background* pada tulisan *Apache* dan *MySQL* akan berwarna hijau yang mengindikasikan bahwa aplikasi sudah terhubung ke *database* secara langsung, seperti yang dapat dilihat pada gambar 4.



Gambar 4. Control panel XAMPP setelah terhubung ke *database*

4. Setelah terhubung ke *database*, jalankan aplikasi alat ukur *database*. Jika sudah dijalankan akan tampak login GUI (*Graphical User Interface*) seperti tampak pada gambar 5.



Gambar 5. Login Interface

5. Masukkan *username* dan *password* lalu *login* ke aplikasi tersebut.
6. Setelah *login*, akan ada tampilan *home* aplikasi. Di *interface* ini pengguna dapat pencarian *database* balita berdasarkan informasi terkait yang dimasukkan. Jika ingin membuat data baru untuk melakukan pengukuran balita yang baru pula, masukkan informasi secara lengkap pada biodata yang tertera secara lengkap. Jika sudah, klik *New*. Adapun pilihan *Update* berguna memperbarui informasi yang salah atau informasi terbaru dari salah satu balita, *Clear* untuk membersihkan semua kolom biodata yang tertera, dan *Delete* untuk

menghapus salah satu data yang ingin dihapus. Tampilan *Input data interface* dapat dilihat pada gambar 6.



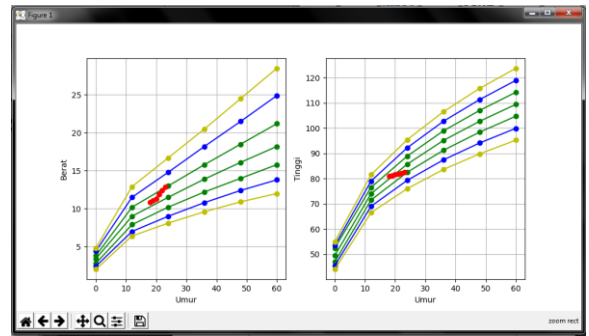
Gambar 6. *Input data Interface*

7. Jika sudah memilih balita mana yang ingin diukur, klik *Measurement*. Tampilan GUI akan berubah seperti yang terlihat pada gambar 7.



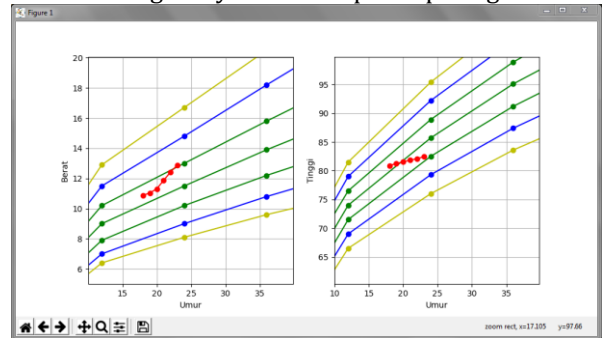
Gambar 7. *Measurement Interface*

8. Untuk mengukur, isi terlebih dahulu kolom pengukuran ke-berapakah yang sedang dilakukan dan tanggal pengukuran.
9. Jika sudah, klik *Measure* dimana aplikasi tersebut akan mulai menghitung pengukuran *realtime* dari *hardware* alat ukur APBT yang akan ditampilkan pada kolom Tinggi dengan satuan cm dan Berat dengan satuan kg. Karena sifat sensor ultrasonik pada alat bersifat *endless* atau terus mengukur tanpa henti, tombol *Get* perlu di-klik agar pengukuran berhenti. Meskipun menggunakan tombol *Get*, hasil ukurnya terbilang cukup akurat.
10. Apabila pengukuran telah dilakukan dan didapat hasil ukurnya, klik tombol *Save* agar hasil ukur berdasarkan biodata terkait disimpan ke *database*.
11. Pilih hasil ukur yang tertera pada tabel jika ingin melihat lagi hasil ukur yang telah dilakukan sebelumnya. Klik *Delete* jika ingin menghapus hasil ukur yang salah atau tidak diinginkan dan klik *Edit* jika ingin memperbaiki data tertentu dari hasil ukur tersebut.
12. Klik *Check Graphic* jika ingin melihat hasil ukur balita dalam bentuk grafik dan melihat pertumbuhannya normal atau tidak. Adapun bentuk grafik yang akan ditampilkan adalah seperti pada gambar 8 berikut.



Gambar 8. Grafik pertumbuhan balita

Adapun tampilan grafik jika sudah di-zoom pada kendali navigasi nya akan tampak seperti gambar 9.



Gambar 9. Grafik pertumbuhan balita jika di-zoom

13. Klik *Convert To Excel* untuk mengubah format *database* menjadi *excel*. Ini dilakukan jika pengguna aplikasi ingin menggunakan data tadi untuk rekapitulasi data apabila ingin dicetak nantinya. Adapun bentuk *file excel* yang akan dihasilkan adalah pada gambar 10 berikut.

	A	B	C	D	E
1	nama	pengukuran	tanggal	berat	tinggi
2	Khairunisa Salsabila Putri	1	4 Agustus 2020	10.58	79.26
3	Khairunisa Salsabila Putri	2	11 Agustus 2020	10.66	79.77
4	Khairunisa Salsabila Putri	3	18 Agustus 2020	10.52	79.82
5					

Gambar 10. Data perkembangan balita yang dikonversikan menjadi file excel

B. Pembahasan Program pada APBT

Adapun bagian-bagian program yang akan digunakan untuk dibahas adalah sebagai berikut.

1. Inisialisasi macam-macam *modules* yang akan digunakan.
Libraries atau *modules* yang akan diinisialisasi adalah *libraries* yang hanya dibutuhkan untuk aplikasi tersebut, diantaranya adalah:
 - a. *Line 1* sampai *line 4* merupakan tampilan GUI, pemberitahuan *form* biodata hingga pesan pemberitahuan seperti *error*.
 - b. *Line 5* merupakan modul untuk akses ke *database*, komunikasi *serial*, dan akses untuk mengkonversikan *file* menjadi *.xlsx (file excel)*
 - c. *Line 6* juga merupakan akses komunikasi *serial*

- d. Line 7 adalah modul untuk menghasilkan tampilan grafik data nantinya
- e. Line 8 merupakan modul untuk pengolahan data yang tak beraturan menjadi beraturan dan bisa diolah (*dataframe*)

```

1  from tkinter import *
2  from tkinter import ttk
3  from tkinter import messagebox
4  from tkinter import filedialog
5  import pymysql, serial, openpyxl
6  import serial.tools.list_ports
7  import matplotlib.pyplot as plt
8  import pandas as pd
9

```

Gambar 11. Inisialisasi *modules*

2. Inisialisasi *function* atau *method*

Function atau *Method* adalah program yang dibuat untuk menjalankan suatu tugas tertentu. Dengan menggunakan *function*, program bisa digunakan berkali-kali hanya dengan mendefinisikan fungsi sesuai kebutuhan program itu sendiri. Sehingga akan lebih efisien untuk pembuatan aplikasinya. Adapun macam-macam fungsi yang dibuat adalah sebagai berikut:

- a. Fungsi untuk membuat tampilan GUI berubah-ubah.

```

10 def raise_frame(frame):
11     frame.tkraise()
12

```

Gambar 12. Fungsi untuk membuat tampilan GUI berubah-ubah

- b. Fungsi keamanan *login* agar aplikasi tidak mudah digunakan sembarangan orang. Adapun *username* dan *password* nya bisa diubah terlebih dahulu sebelum aplikasi benar-benar akan digunakan.

```

13 def login_system():
14     u = (Username.get())
15     p = (Password.get())
16     if (u == ('admin') and p == ('password')):
17         raise_frame(Window2)
18     else:
19         messagebox.showinfo('Login Systems', 'Akses Ditolak!!\nCoba Lagi!!!')
20         Reset()
21

```

Gambar 13. Fungsi keamanan login pada aplikasi

- c. Fungsi tombol *reset* pada *login* page.

```

22 def Reset():
23     Username.set("")
24     Password.set("")
25     txtUsername.focus()
26

```

Gambar 14. Fungsi tombol *reset* pada *login* page

- d. Fungsi tombol *exit* pada *login* page

```

27 def iExit():
28     iExit = tkinter.messagebox.showinfo("Login Systems", "You decided to Exit!!!")
29     root.destroy()
30

```

Gambar 15. Fungsi tombol *exit* pada *login* page

- e. Fungsi akses komunikasi *serial* yang terhubung pada *desktop*

```

31 def serial_ports():
32     return serial.tools.list_ports.comports()
33

```

Gambar 16. Fungsi akses komunikasi *serial*

- f. Fungsi akses pilihan komunikasi *serial* yang tersedia

```

34 def on_select(event = None):
35     print('event widgets : ', event.widget.get())
36     print('comboBox : ', portCombo.get())
37

```

Gambar 17. Fungsi akses pilihan komunikasi *serial* yang tersedia

- g. Fungsi penyesuaian data pada *database* setelah dilakukan suatu perubahan. Perubahan yang dimaksud seperti penambahan, pengurangan, perubahan, dan lain-lain.

```

38 def fetch_data():
39     con = pymysql.connect(host = 'localhost', user = 'root', password = '', database = 'stm')
40     cur = con.cursor()
41     cur.execute("SELECT * from baby WHERE id = %s, nama = %s", ID_vr.get(), Nama.get())
42     rows = cur.fetchall()
43     if len(rows) != 0:
44         tabelUkur.delete(tabelUkur.get_children())
45         for row in rows:
46             tabelUkur.insert('', END, values = row)
47         con.commit()
48     con.close()
49

```

Gambar 18. Fungsi penyesuaian data pada *database*

- h. Fungsi penghentian pengukuran data terus-menerus

```

50 def pause():
51     root.after_cancel(update)
52

```

Gambar 19. Fungsi penghentian pengukuran data terus-menerus.

- i. Fungsi pembacaan nilai data dari *Arduino* ke *desktop*

```

57 def nilaiSensor():
58     if True:
59         global nilaiUkurTinggi, nilaiUkurBerat, root, update
60         x = arduinoSerialData.readline()
61         values = str(x.decode().strip())
62         y = values.split('#')
63         Tinggi.set(y[0])
64         Berat.set(y[1])
65         update = root.after(700, nilaiSensor)
66

```

Gambar 20. Fungsi pembacaan nilai data

j. Fungsi tombol *add* pada *input data interface*

```

136 def add_balita():
137     if ID_vr.get() == '' or Nama.get() == '' or Nama_Ibu.get() == '':
138         messagebox.showerror('Error', 'Semua Data Harus Disiisi Terlebih Dahulu!!')
139     else:
140         con = pymysql.connect(host = 'localhost', user = 'root', password = '', database = 'stm')
141         cur = con.cursor()
142         cur.execute("INSERT INTO baby VALUES(%s, %s, %s, %s, %s, %s, %s, %s)", (ID_vr.get(),
143             Nama.get(),
144             Jenis_Kelamin.get(),
145             Tempat_Lahir.get(),
146             Tanggal_Lahir.get(),
147             Anak_Ke.get(),
148             Nama_Ayah.get(),
149             Nama_Ibu.get(),
150             txt_Alamat.get('1.0', END)
151         ))
152         con.commit()
153         fetch_data()
154         con.close()
155         messagebox.showinfo('Success', 'Biodata Telah Disimpan!!')
    
```

Gambar 21. Fungsi tombol *add* pada *input data interface*

k. Fungsi tombol *clear* pada *input data interfaces*

```

171 def clear():
172     ID_vr.set('')
173     Nama.set('')
174     Jenis_Kelamin.set('')
175     Tempat_Lahir.set('')
176     Tanggal_Lahir.set('')
177     Anak_Ke.set('')
178     Nama_Ayah.set('')
179     Nama_Ibu.set('')
180     txt_Alamat.delete('1.0', END)
181
    
```

Gambar 22. Fungsi tombol *clear* pada *input data interface*

l. Fungsi menampilkan biodata pada *form*

```

182 def get_cursor(ev):
183     cursor_row = baby_Table.focus()
184     contents = baby_Table.item(cursor_row)
185     row = contents['values']
186     ID_vr.set(row[0])
187     Nama.set(row[1])
188     Jenis_Kelamin.set(row[2])
189     Tempat_Lahir.set(row[3])
190     Tanggal_Lahir.set(row[4])
191     Anak_Ke.set(row[5])
192     Nama_Ayah.set(row[6])
193     Nama_Ibu.set(row[7])
194     txt_Alamat.delete('1.0', END)
195     txt_Alamat.insert(END, row[8])
196
    
```

Gambar 23. Fungsi menampilkan biodata pada *form*

m. Fungsi memperbarui biodata

```

197 def update_data():
198     con = pymysql.connect(host = 'localhost', user = 'root', password = '', database = 'stm')
199     cur = con.cursor()
200     cur.execute(
201         "UPDATE baby SET nama = %s, kelamin = %s, tempat = %s, tanggal = %s, anak = %s, ayah = %s, ibu = %s, alamat = %s WHERE id = %s",
202         (
203             Nama.get(),
204             Jenis_Kelamin.get(),
205             Tempat_Lahir.get(),
206             Tanggal_Lahir.get(),
207             Anak_Ke.get(),
208             Nama_Ayah.get(),
209             Nama_Ibu.get(),
210             txt_Alamat.get('1.0', END),
211             ID_vr.get()
212         ))
213     con.commit()
214     fetch_data()
215     con.close()
216     messagebox.showinfo('Success', 'Biodata Telah Diupdate!!')
217
    
```

Gambar 24. Fungsi memperbarui biodata yang ada pada *database*

n. Fungsi menghapus biodata

```

218 def delete_data():
219     con = pymysql.connect(host = 'localhost', user = 'root', password = '', database = 'stm')
220     cur = con.cursor()
221     cur.execute("DELETE FROM baby WHERE id = %s", ID_vr.get())
222     con.commit()
223     con.close()
224     fetch_data()
225     clear()
226
    
```

Gambar 25. Fungsi menghapus biodata

o. Fungsi menampilkan grafik pertumbuhan balita. Data pertumbuhan diambil dari *database* lalu diolah agar bisa terbaca untuk ditampilkan secara grafik

```

319 def check_graphics():
320     con = pymysql.connect(host = 'localhost', user = 'root', password = '', database = 'stm')
321     cur = con.cursor()
322     cur.execute("SELECT * FROM graph ORDER BY graph . tanggal ASC")
323     cur.execute("SELECT tanggal, berat, tinggi FROM graph WHERE id = {}".format(ID_vr.get()))
324
325     query = "SELECT * FROM graph WHERE id = {}".format(ID_vr.get())
326     read_SQL = pd.read_sql_query(query, con)
327     df = pd.DataFrame(read_SQL, columns = ['nama', 'pengukuran', 'tanggal', 'berat', 'tinggi'])
328
329     query1 = "SELECT * FROM baby WHERE id = {}".format(ID_vr.get())
330     read_SQL1 = pd.read_sql_query(query1, con)
331     df1 = pd.DataFrame(read_SQL1, columns = ['nama', 'kelamin', 'tanggal'])
332
333     gender = df1['kelamin'].tail(1).str.split().tolist()
334     gender_1 = gender[0][0]
335     print(gender_1)
336     print(type(gender_1))
337
338     result = cur.fetchall()
339     tanggal = []
340     berat = []
341     berat_new = []
342     tinggi = []
343     tinggi_new = []
344
    
```

Gambar 26. Fungsi menampilkan grafik pertumbuhan balita

```

345 for record in result:
346     tinggi.append(record[2])
347     berat.append(record[1])
348     tanggal.append(record[0])
349
350 for ax in range(len(berat)):
351     new_ax = float(berat[ax])
352     berat_new.append(new_ax)
353     ax += 1
354
355 for bx in range(len(tinggi)):
356     new_bx = float(tinggi[bx])
357     tinggi_new.append(new_bx)
358     bx += 1
    
```

Gambar 27. Fungsi menampilkan grafik pertumbuhan balita (lanjutan pertama)

```

365 beratUkur = berat_new[-1]
366 tinggiUkur = tinggi_new[-1]
367
368 case = {
369     'Januari' : 1,
370     'Februari' : 2,
371     'Maret' : 3,
372     'April' : 4,
373     'Mei' : 5,
374     'Juni' : 6,
375     'Juli' : 7,
376     'Agustus' : 8,
377     'September' : 9,
378     'Oktober' : 10,
379     'November' : 11,
380     'Desember' : 12
381 }
382
383 bulanLahir = df1['tanggal'].tail(1).str.split().tolist()
384 bulanLahir_1 = bulanLahir[0][1]
385 bulanLahir_2 = case.get(bulanLahir_1)
386 tahunLahir_1 = int(bulanLahir[0][2])
387
388 bulanUkur = df['tanggal'].str.split().tolist()
    
```

Gambar 28. Fungsi menampilkan grafik pertumbuhan balita (lanjutan kedua)

```

390 m = 0
391 umurUkur = []
392 for umurBulan in bulanUkur:
393     umurBulan = bulanUkur[m][1]
394     tahunUkur = int(bulanUkur[m][-1])
395     umurBulan_1 = case.get(umurBulan)
396     umurDalamBulan = umurBulan_1 - bulanLahir_2
397     totalUmur = ((tahunUkur - tahunLahir_1) * 12) + umurDalamBulan
398     umurUkur.append(totalUmur)
399     m += 1
400
401 bulanUkur = df['tanggal'].tail(1).str.split().tolist()
402 bulanUkur_1 = bulanUkur[0][1]
403 bulanUkur_2 = case.get(bulanUkur_1)
404 tahunUkur_1 = int(bulanUkur[0][2])
405
406 batasUmur = [0, 12, 24, 36, 48, 60]

```

Gambar 29. Fungsi menampilkan grafik pertumbuhan balita (lanjutan ketiga)

```

408 # RMS laki-laki
409
410 beratLakiBurukOver = [5.0, 13.3, 17.0, 20.6, 23.8, 27.3]
411 tinggiLakiBurukOver = [56.0, 83.0, 96.3, 107.2, 115.9, 123.7]
412
413 beratLakiCukupBurukOver = [4.5, 12.0, 15.3, 18.4, 21.2, 24.1]
414 tinggiLakiCukupBurukOver = [54.0, 80.5, 93.2, 103.5, 111.7, 119.2]
415
416 beratLakiCukupNormalOver = [3.9, 10.7, 13.7, 16.3, 18.5, 21.0]
417 tinggiLakiCukupNormalOver = [52.0, 78.0, 90.2, 99.8, 107.5, 114.6]
418
419 beratLakiNormal = [3.4, 9.6, 12.3, 14.3, 16.3, 18.3]
420 tinggiLakiNormal = [50.0, 75.5, 87.1, 96.1, 103.3, 110.0]
421
422 beratLakiCukupNormalUnder = [2.9, 8.6, 10.8, 12.7, 14.5, 16.0]
423 tinggiLakiCukupNormalUnder = [48.0, 73.5, 84.1, 92.4, 99.1, 105.3]
424
425 beratLakiCukupBurukUnder = [2.5, 7.7, 9.7, 11.4, 12.7, 14.1]
426 tinggiLakiCukupBurukUnder = [46.0, 71.0, 81.0, 88.7, 94.9, 100.7]
427
428 beratLakiBurukUnder = [2.1, 6.9, 8.6, 10.0, 11.4, 12.5]
429 tinggiLakiBurukUnder = [44.0, 68.5, 78.0, 85.0, 90.7, 96.1]
430

```

Gambar 30. Fungsi menampilkan grafik pertumbuhan balita (lanjutan keempat)

```

432 # RMS perempuan
433
434 beratPerempuanBurukOver = [4.8, 12.9, 16.7, 20.5, 24.5, 28.5]
435 tinggiPerempuanBurukOver = [55.0, 81.5, 95.4, 106.5, 115.7, 123.7]
436
437 beratPerempuanCukupBurukOver = [4.4, 11.5, 14.8, 18.2, 21.5, 24.9]
438 tinggiPerempuanCukupBurukOver = [53.5, 79.0, 92.2, 102.7, 111.3, 118.9]
439
440 beratPerempuanCukupNormalOver = [3.8, 10.2, 13.0, 15.8, 18.5, 21.2]
441 tinggiPerempuanCukupNormalOver = [52.5, 76.5, 88.9, 98.9, 107.0, 114.2]
442
443 beratPerempuanNormal = [3.4, 9.0, 11.5, 13.9, 16.1, 18.2]
444 tinggiPerempuanNormal = [49.5, 74.0, 85.7, 95.1, 102.7, 109.4]
445
446 beratPerempuanCukupNormalUnder = [2.8, 7.9, 10.2, 12.2, 14.0, 15.8]
447 tinggiPerempuanCukupNormalUnder = [47.0, 71.5, 82.5, 91.2, 98.4, 104.7]
448
449 beratPerempuanCukupBurukUnder = [2.4, 7.0, 9.0, 10.8, 12.4, 13.8]
450 tinggiPerempuanCukupBurukUnder = [45.5, 69.0, 79.3, 87.4, 94.1, 99.9]
451
452 beratPerempuanBurukUnder = [2.0, 6.4, 8.1, 9.6, 10.9, 12.0]
453 tinggiPerempuanBurukUnder = [44.0, 66.5, 76.0, 83.6, 89.8, 95.2]
454
455 fig, (ax1, ax2) = plt.subplots(1, 2, figsize = (10, 5))

```

Gambar 31. Fungsi menampilkan grafik pertumbuhan balita (lanjutan kelima)

```

832 ax1.plot(batasUmur, beratPerempuanNormal, 'g-o')
833 ax1.plot(batasUmur, beratPerempuanCukupNormalOver, 'g-o')
834 ax1.plot(batasUmur, beratPerempuanCukupNormalUnder, 'g-o')
835 ax1.plot(batasUmur, beratPerempuanCukupBurukOver, 'b-o')
836 ax1.plot(batasUmur, beratPerempuanCukupBurukUnder, 'b-o')
837 ax1.plot(batasUmur, beratPerempuanBurukOver, 'y-o')
838 ax1.plot(batasUmur, beratPerempuanBurukUnder, 'y-o')
839 ax1.plot(umurUkur, berat_new, 'r-o')
840
841 ax2.plot(batasUmur, tinggiPerempuanNormal, 'g-o')
842 ax2.plot(batasUmur, tinggiPerempuanCukupNormalOver, 'g-o')
843 ax2.plot(batasUmur, tinggiPerempuanCukupNormalUnder, 'g-o')
844 ax2.plot(batasUmur, tinggiPerempuanCukupBurukOver, 'b-o')
845 ax2.plot(batasUmur, tinggiPerempuanCukupBurukUnder, 'b-o')
846 ax2.plot(batasUmur, tinggiPerempuanBurukOver, 'y-o')
847 ax2.plot(batasUmur, tinggiPerempuanBurukUnder, 'y-o')
848 ax2.plot(umurUkur, tinggi_new, 'r-o')
849
850 ax1.set_xlabel('Umur')
851 ax1.set_ylabel('Berat')
852 ax1.grid()
853
854 ax2.set_xlabel('Umur')
855 ax2.set_ylabel('Tinggi')
856 ax2.grid()
857
858 plt.show()

```

Gambar 32. Fungsi menampilkan grafik pertumbuhan balita (lanjutan keenam)

p. Fungsi mengkonversi *database* menjadi *file excel (.xlsx)*

```

861 def convert_data():
862     conn = pymysql.connect(host = 'localhost', user = 'root', password = '', database = 'stm')
863     query = "SELECT * FROM graph WHERE id = {}".format(ID_vr.get())
864     read_sql = pd.read_sql_query(query, conn)
865     df = pd.DataFrame(read_sql, columns = ['nama', 'pengukuran', 'tanggal', 'berat', 'tinggi'])
866
867     file_path = filedialog.asksaveasfilename(defaultextension = '.xlsx')
868     df.to_excel(file_path, index = False, header = True)
869     conn.close()

```

Gambar 33. Fungsi mengkonversikan *database* menjadi *file excel*

IV. PENUTUP

Aplikasi yang diuji telah bekerja dengan baik sesuai dengan rancangan yang telah dilakukan sebelumnya. Data yang tersimpan akurat, sesuai dengan alat (*hardware*) yang mengukur hasil pengukurannya. *Database* dapat disimpan dengan baik dan aman, serta bisa digunakan kembali sesuai dengan metode yang disebutkan sebelumnya, yakni *Realtime Active Database Management System*.

REFERENSI

- [1] R. Patrio Timi, "Rancang Bangun Pengukur Berat Dan Tinggi Badan Balita Otomatis Berbasis Mikrokontroler", Rancang Bangun Pengukur Berat Dan Tinggi Badan Balita Otomatis Berbasis Mikrokontroler, 2019.
- [2] S. Anita, and K. S. Devendra, "Real Time Application of Database Management System Using Monitoring of Input", International Journal of Advanced Engineering Research and Science (IJAERS), 1(3), 8-11.
- [3] Juli Sardi, Habibullah, Risfendra, " Rancang Bangun Sistem Monitoring Pertumbuhan dan Berat Badan Balita Berbasis Data pada Posyandu", Jurnal Elkha, Vol.11 No.2, 2019.

- [4] Bella Hardiyana dan Irfan Suendi, "Sistem Informasi Pendaftaran Bayi," Jurnal Teknologi dan Informasi, Vol 1, no 3, 2013.
- [5] Nabila Sholiha dan Sri Kusamadewi, "Sistem Informasi Posyandu Kesehatan Ibu dan Anak," Prosiding Seminar Nasional dan Informatika Ke 2, 2015.
- [6] Indrajani, "Membangun Basis Data Posyandu Menuju Indonesia Sehat Sentosa," Jurnal Comtech, Vol. 4, no.2, pp.618-626, 2013.
- [7] Nofriadi dan Alpin Aperta, "Perancangan Aplikasi Timbangan Bayi pada Posyandu dengan Standar Antropometri WHO 2005 Menggunakan Arduino Uno R3, Ms. Visual Studio.Net 2010 dan My SQL," Jurnal CoreIT, Vol.3, No.1, pp. 1-8, Juni.2017
- [8] Wen Hao, Dong Xiao-rui, Ma Yu-cheng and Nan Jin-rui, "The Research of the Databases Connection Methods in Lab View Based on ADO", International Conference on Computer Application and System Modeling (ICCA SM), 2010.
- [9] Xuejie Wei, Jie Zhang, Zhao Yang and Zhen Zhao, "The Management System for Data Acquisition based on LabView and LabSQL", Third International Conference on Information Science and Technology, March 23-25, 2013.
- [10] Song qiang, Shi Yanfang and Lv Chenguang, " Database Design in Data Acquisition System for Electric Vehicle's Driving Motor Test Bench," Fifth Conference on Measuring Technology and Mechatronics Automation, 2013.

Biodata Penulis

Juli Sardi, lahir di Pulau Punjung, 18 Juli 1987. Sarjana Pendidikan di Jurusan Teknik Elektro FT UNP 2010. Tahun 2003 memperoleh gelar Master Teknik di jurusan Teknik Elektro Pasca sarjana ITS. Staf pengajar di jurusan Teknik Elektro FT UNP sejak tahun 2014- sekarang.