

**Rolly Maulana Awangga**

**Jose Chasey Pratama**

**Christian Yuda Pratama**

**Flutter dan Python**

Tantangan Membangun Aplikasi dengan Flutter dan Python

**Rolly Maulana Awangga**

**Jose Chasey Pratama**

**Christian Yuda Pratama**



**Flutter dan Python**

Tantangan Membangun Aplikasi flutter dan python

***Penulis :***

Jose Chasey Pratama

Christian Yuda Pratama

Rolly Maulana Awangga

ISBN : -

***Editor :***

Roni Andarsyah

***Penyunting :***

Roni Andarsyah

***Desain sampul dan Tata letak :***

Idam Fadilah

***Penerbit :***

Penerbit Buku Pedia

***Redaksi :***

Athena Residence Blok. E No. 1, Desa Ciwaruga,

Kec. Parongpong, Kab. Bandung Barat 40559

Tel. 628-775-2000-300

Email : penerbit@bukupedia.co.id

***Distributor :***

Informatics Research Center

Jl. Sariasih No. 54

Bandung 40151

Email : irc@poltekpos.ac.id

Cetakan Pertama, 2022

Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk dan

dengan cara apa pun tanpa ijin tertulis dari penerbit

**KATA PENGANTAR**

Dengan hormat,

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa, atas rahmat dan karunia-Nya sehingga buku ini dapat terselesaikan. Buku ini ditulis dengan harapan dapat memberikan manfaat bagi para developer dan pemula dalam mempelajari Flutter dan Python, Sampai saat ini pembuatan aplikasi berbasis android sangat banyak digemari dan digunakan pada era digital saat ini.

Kami berharap buku ini dapat memberikan panduan yang berguna bagi para developer dan pemula yang ingin membuat aplikasi dengan menggunakan python dan flutter. Kritik dan saran yang membangun sangat kami harapkan demi kesempurnaan buku ini.

Bandung, TGL/ BLN 2023

Penulis

**DAFTAR ISI**

**KATA PENGANTAR i**

**PRAKATA i**

**DAFTAR ISI i**

**BAB 1 JUDUL BAB 1**

1. Pendahuluan 1
2. Tujuan Instruksional dan Capaian Pembelajaran 1
3. Uraian materi 1
4. Latihan 1
5. Rangkuman materi 1
6. Pustaka 1

**BAB 2 JUDUL BAB 1**

1. Pendahuluan 1
2. Tujuan Instruksional dan Capaian Pembelajaran 1
3. Uraian materi 1
4. Latihan 1
5. Rangkuman materi 1
6. Pustaka 1

**GLOSARIUM**

**INDEKS 1**

**PROFIL PENULIS 1**

**BAB 1**

**APA ITU FLUTTER? DAN PYTHON?**

1. **PENDAHULUAN**

Flutter merupakan sebuah framework bersifat open-source yang dikenalkan oleh google sejak juni 2018, dimana berguna sebagai pembangunan antarmuka (user interface/UI) aplikasi untuk Android dan iOS. Flutter ditulis dengan menggunakan Bahasa Dart . Pada era Transformasi digital saat ini, ingin semua kegiatan menjadi lebih cepat dan efisien karena dibantu oleh adanya teknologi. Peminatan flutter sebagai Bahasa pemrograman yang semakin populer dikalangan frontend. Tak terlepas juga dalam pembuatan sebuah server dibutuhkan kekuatan yang cepat dan tidak memakan resource yang banyak. Python menjadi salah satu konsumsi bagi para programmer backend. Python merupakan sebuah Bahasa yang interpretative dimana python dianggap mudah untuk dipelajari serta berfokus pada keterbacaan sebuah kode. Dengan kata lain, python diklaim sebagai Bahasa pemrograman yang memiliki masing – masing kode yang jelas ,lengkap dan efisien. Namun pada kenyataannya sering kali terjadi kesulitan dalam membangun aplikasi yang responsive dengan membangun antarmuka tersebut kemudian juga lambatnya server yang dibutuhkan suatu pengembang.

Oleh karena itu pembuatan sebuah aplikasi dengan menggunakan flutter dan python adalah jawaban yang tepat dimana flutter sebagai pembangun antarmuka yang stabil dan responsive bagi pengguna Android dan iOS serta python yang mampu meminimalisir sebuah resource bagi server tentunya akan lebih cepat dalam pemilihan sebagai server. Harapannya dengan dibuatkan buku ini pembaca dapat mempelajari cara membuat / membangun sebuah aplikasi dengan flutter dan python.

## **1.1 Pengenalan Flutter**

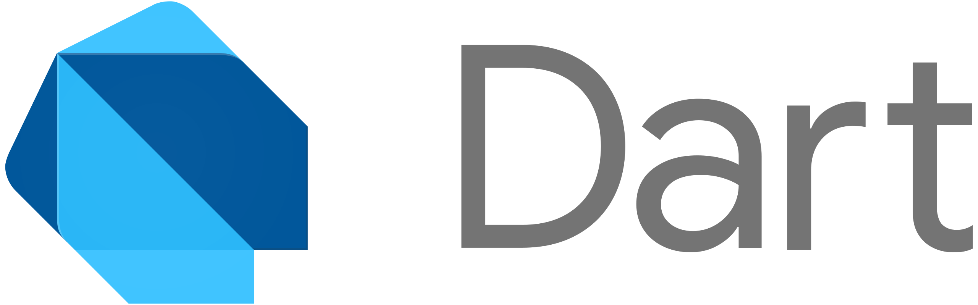
Sebelum masuk dan memulai untuk mendekonstruksikan flutter , mari kita akan pelajari dan sedikit mengenal bahasa yang akan digunakan untuk membuatnya yaitu *Dart*. *Dart* merupakan pemrograman beroirientasi objek yang diperkenalkan oleh *Google* untuk pertama kalinya pada tahun 2011. Semenjak saat itu Dart terus berkembang dengan merilis fitur – fitur yang sangat memungkinkan kompilasi untuk *platform* *Windows, Linux dan macOS*  sebagai aplikasi desktop.

Secara umum, bahasa Dart mudah untuk dipelajari khususnya bagi para pengembang atau pun bagi para pemula. Dikearenakan Sintaksnya yang dipakai sangat mirip dengan bahasa *java*, *swift* ataupun *kotlin*. Selain itu, kit pengembangan perangkat lunak *Dart* atau *SDK* dikirimkan dengan sebuah Dart *Virtual* *Machine* ( VM ) yang beridiri sendiri tentunya akan sangat memungkinkan anda dimana membuat kode dilingkungan antarmuka baris perintah ( CLI ) dan jika anda tidak membiasakan pada perintah *CLI* tentunya anda dapat bermain – main dengan *Dart* di *DartPad*.

*DartPad* adalah editor online dimana menyediakan sebuah akses ke *API* D*art* dan dapat memungkinkan kita untuk mengkompilasi sebuah kode *Dart*, jika kita ingin mempelajari bahasa *Dart*, tidaklah sulit jika sudah mempunyai pengalaman dengan *Java* ataupun *JavaScript*

Dart adalah sebuah bahasa yang sudah dioptimalkan untuk user demi mengembangkan aplikasi cepat pada *platform* apa pun. Tujuannya adalah untuk menawarkan bahasa pemrograman yang paling produktif untuk melakukan pengembangan multi-platform, dipasangkan dengan *platform* *runtime* eksekusi yang fleksibel dalam kerangka kerja aplikasi.

*Dart* adalah sebuah bahasa yang sudah dioptimalkan untuk user demi mengembangkan aplikasi cepat pada platform apa pun. Tujuannya adalah untuk menawarkan bahasa pemrograman yang paling produktif untuk melakukan pengembangan multi-platform, dipasangkan dengan platform runtime eksekusi yang fleksibel dalam kerangka kerja aplikasi.



Bahasa ditentukan oleh technical envelope – dimana pilihan yang dibuat oleh pengembangan dalam melakukan pembentukan kemampuan dan kekuatan bahasa. Sebab itu *Dart* dirancang untuk lingkup teknis yang sangat cocok untuk melakukan pengembangan klien, memprioritaskan pengembangan dan pengalaman produksi yang berkualitas tinggi dan di berbagai target kompilasi (*web*, *desktop*, dan *mobile* ). Dart juga membentuk dasar Flutter. Dimana dart menyediakan bahasanya dan waktu proses yang akan dipakai untuk didukung dasar *Flutter*.

Seperti yang diketahui flutter adalah framework aplikasi *mobile* yang diciptakan oleh *google*. pengembangan aplikasi untuk sistem operasi seperti *Android*, *iOS*, *Windows*, *Linux, MacOS*, serta menjadikan metode utama dalam membuat aplikasi *Google Fuchsia*. Versi Pertama dari flutter dikenal sebagai “*Sky*” dan berjaaln pada sistem operasi *Android*. Diresmikan pada perhelatan *Dart developer summit* tahun 2015, dengan tujuan untuk mampu merender grafis secara konsisten pada 120 bingkai perdetik.

Akan tetapi dart juga sangat mendukung banyak tugas pengembang inti seperti menganalisis, menformat ,menguji sebuah kode. Bahasa Dart adalah tipe aman; ia menggunakan pemerikasaan sebuah tipe statis untuk memastikan tentunya dimana nilai variable selalu cocok dengan tipe statis variable. Terkadang hal ini juga disebut sebagai pengetikan suara. Meskipun tipe bersifat wajib, anotasi tipe bersifat opsional karena inferensi tipe.

Sistem pengetikan dart juga fleksibel, memungkinkan dimana penggunaannya yang dikombinasikan dengan pemeriksaan *runtime*, yang dapat berguna selama eksperimen atau untuk kode yang perlu sangat dinamis.

Dart menawarkan keamanan yaitu *null* yang baik, dimana nilai tidak boleh *null* kecuali anda mengatakannya bisa. Dengan keamanan *null*, *Dart* dapat melindungi tentunya dari pengecualian *null* saat *runtime* melalui bentuk analisis kode statis. Tidak seperti banyak bahasa yang bersifat *null*-*safe* lainnya, Ketika *Dart* menentukan bahwasannya suatu variable tentu tidak dapat dibatalkan, variabel itu selalu tidak dapat dibatalkan,



(Sumberhttps://cdn-images-1.medium.com/max/1024/1\*6-SfhIoFLl1NPaNi8Ld8Og.png)

Flutter 1.0 dirilis pada tanggal 4 Desember 2018 di acara Flutter *Live* yang ditunjukkan sebuah versi stabil pertama dari *framework* *Flutter*.

Jika Anda memeriksa kode yang sedang berjalan didebugger, anda akan melihat bahwa *non*-*nullability* dipertahankan pada saat *runtime*(oleh karenanya tidak terdengar aman).

Sebuah Teknologi *compiler* dari *Dart* memungkinkan Anda menjalankan kode dengan cara yang berbeda:

1. *Platform* *web*: sebagai aplikasi yang akan menargetkan *web*, *Dart* menyertakan kompiler waktu dalam pengembangan (*dartdevc*) serta kompiler waktu produksi(*dart2js*) dimana kedua kompiler tersebut menerjemahkan Dart kedalam *JavaScript*.
2. *Platform* *native*: sebagai aplikasi dimana menargetkan perangkat seluler dan desktop, *Dart* menyertakan VM *Dart* dengan sebuah kompilasi *just*-*in*-*time* ( JIT ) dan compiler sebelumnya (AOT) untuk memproduksikan kode mesin.

Memiliki 2 komponen utama yaitu :

* *Dart Platform*

Flutter ditulis dengan Bahasa pemograman Dart. Dimana *flutter* sendiri berjalan pada mesin virtual Dart yang dilengkapi mesin eksekusi kompilasi tepat waktu ( *just* in *time* ). Flutter menggunakan kompilasi tepat waktu untuk melakukan “hot reload”, yang dapat menambahkan hasil modifikasi kode langsung kedalam aplikasi yang sedang dijalankan.

* *Flutter Engine*

Flutter engine yaitu memberikan dukungan pada rendering tingkat rendah menggunakan *library* grafik Skia oleh *Google* ditulis dengan menggunakan Bahasa pemrograman *C++.* Flutter engine dapat berinteraksi dengan perkakaks pengembangan perangkat lunak atau yang sering disebut sebagai SDK spesifik-serambi (*Platform – Specific*) seperti yang disediakan oleh *Android* dan *iOS.*

* *Foundation Library*
  + 1. *Widget,* desain dalam penggunaan flutter melibatakan sebuah perakitan dengan berbagai *widget.* Didalam flutter sebuah widget diwakili dengan deskripsi yang tetap dari bagian antarmuka penggunanya: semua grafik, termasuk isi teks, adapun bentuk dan animasi dibuat menggunakan sebuah widget. Widget yang lebih kompleks dapat dibuat dengan memadukan widget yang lebih simpel.
    2. *Widget* spesifik-desain, framework pada flutter berisikan dua set widget yang dibentukan dengan Bahasa tertentu. Widget material *design* menerapkan dari Bahasa desain Google dengan nama yang sama, sedangkan widget ‘Cupertino’ meniru desain pada iOS milik Apple.

## **1.2 Cara Kerja Framework Flutter**

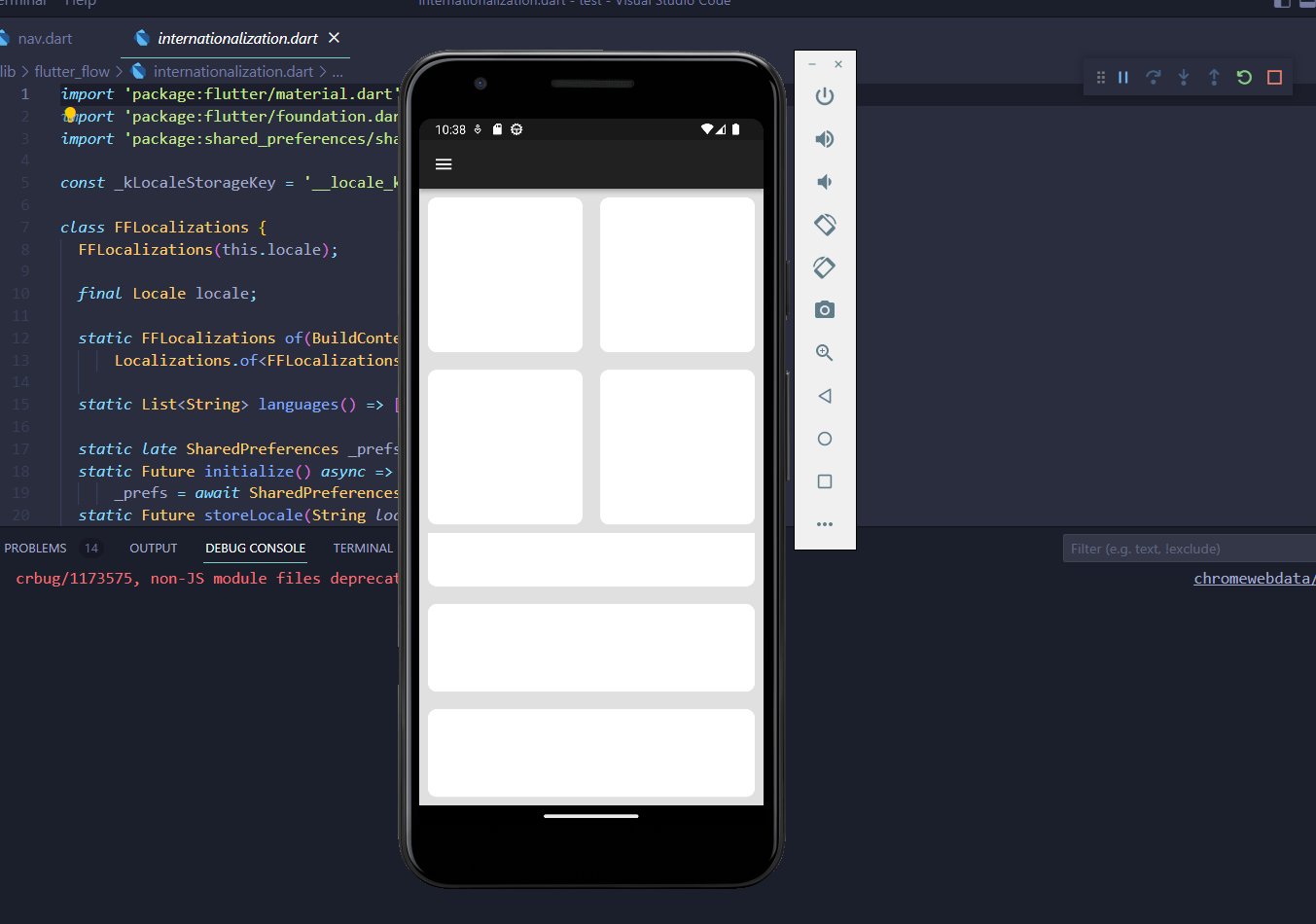
Developer membuat pengembangan aplikasi dengan Bahasa pemograman *Dart*, kemudian widget yang telah tersedia pada *framework* ini. Jika di *Android*, kode yang ditulis akan dikumpulkan dengan tersusun secara teratur dengan mesin C++ menggunakan Android NDK (*Native Development Kit*). Jika pada *iOS*, kode akan distruktur secara teratur dengan mesin kompilasi LLVM (*Low-Level Virtual Machine*). Untuk *Android*, kode native akan distruktur ulang secara teratur dengan Dart Kompiler. Dimana setelah kode disesuaikan dengan masing – masing perangkat, aplikasi bisa dijalankan diberbagai platform.

## **1.3 Kelebihan Flutter**

Berdasarkan penjelasan diatas dapat dikatikan lagi bahwa flutter memiliki kelebihan dan mengapa kita harus menggunakan framework ini? Mari kita simak kelebihan dari flutter. Pengembangan aplikasi dalam mendevelope yang lebih cepat, memiliki alasan tersendiri dengan adanya fitur – fitur yang membantu proses pembuatan atau pengembangan jadi lebih gampang.

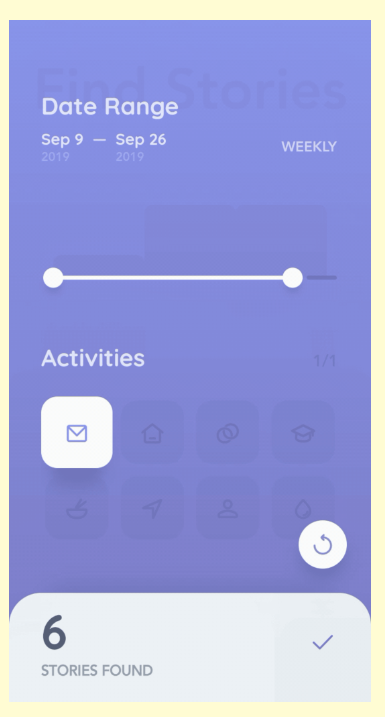
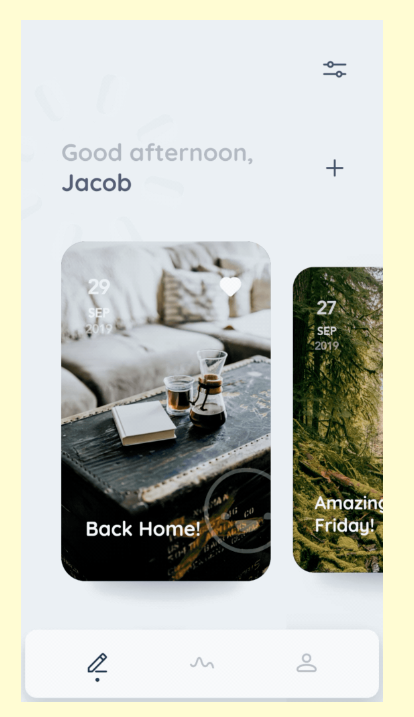
*Flutter* menyediakan sebuah *user interfaces widgets* dengan kualitas coding yang tinggi. *Widget* inin sendiri bisa kita konstruksikan dan kustomisasikan tanpa perlu menulis koding dari nol.

Tak hanya itu fitur *Hot Reload* yang menampilkan kodingan dengan secepat kilat, membuat kita dalam bereksperimen fitur dan memperbaiki sistem bug jadi lebih kilat juga.



Disamping itu, kita tidak perlu menguasi Bahasa lain untuk melakukan pengembangan aplikasi pada *platform* yang beda. Cukup manfaatkan sebuah *widget* yang tersedia oleh *flutter*.

Custom *User Interface* yang menarik, memudahkan developer memuatkan aplikasi, ditawarkannya tampilan user interface yang menari dengan memanfaatkan koleksi widget untuk tampilan aplikasi, seperti *layout*, navigasi, animasi, *style* tema, font bahkan pengaturan tampilan *scrolling*.



*(sumber:* [*2019 Material Design Award Winners - Library - Google Design*](https://design.google/library/material-design-awards-2019/#tripdotcom)*)*

Berikut contoh desain antarmuka seperti yang bisa kita liat dengan menggunakan flutter. Aplikasi reflectly yang memenangkan penghargaan desain aplikasi paling inovatif. Dikarenakan aplikasi ini memiliki sebuah transisi yang *smooth,* animasi yang unik dan gambar yang menarik, dan komposisi warna yang cocok.

## **1.4 Kekurangan Flutter**

Flutter ternyata bukan hanya memiliki kelebihan namun juga memilki kekurangan dalam pengembangannya, ukuran file yang besar masih menjadi salah satu tantangan dari aplikasi yang dikembangkan flutter. Ukuran file yang menjadikanya aplikasi cenderung lebih besar disbanding dengan aplikasi java atau kotlin. Tentu saja dapat merugikan pengguna oleh karena memakan terlalu banyak memori *device*.

Library yang kurang lengkap dari sisi yang dibandingkan dengan native development. Flutter memang masih dianggap baru sehingga semua fungsi yang dibutuhkan oleh developer bisa ditemukan didalam *library,* meski demikian dukungan dari google untuk flutter sangatlah luar biasa sehingga bisa diprediksikan. Disamping itu Flutter masih dianggap memang cukup jauh sehingga perkumpulan flutter pun masih berkembang .

## **1.5 Manfaat Menggunakan Flutter**

Tanpa membuat sebuah perbandingan apapun dengan semua platform lain, berikut adalah beberapa sebuah manfaat, kegunaan dari beberapa sebuah fitur serta kualitas yang mungkin bisa dipertimbangkan untuk pengembang atau pemula mencoba Flutter :

1. **Produktivitas yang Tinggi.** Karena flutter merupakan sebuah lintas platform , kita dapat menggunakan sebuah basis kode yang sama untuk menggunakan aplikasi IOS dan Android kita terlebih ini sangat menghemat waktu dan sumber daya.
2. **Performa yang bagus dan stabil .** Dart mengkompilasikan ke dalam kode aslinya dan tidak perlu mengakses *widget* *OEM* dikarenakan *flutter* yang memilikinya sendiri. Tentunya ini sangat berarti lebih sedikit komunikasi yang akan dimediasikan antara *platform* dan aplikasi. *Flutter* sendiri merupakan satu – satunya *SDK* seluler yang menyediakan sebuah tampilan reaktif tanpa memerlukan jembatan JavaScript. Maka semua ini berkontribusi dengan baik pada waktu startup aplikasi yang cepat dan memasalahkan kinerja yang lebih sedikit.
3. **Perkembangan sederhana dan cepat.** Salah satu yang baik lainnya adalah fitur dalam *Flutter* yang sering dipuji adalah *Hot reload* . Dimana fitur ini sangat berguna bagi para pengembang *android* *developer*  ataupun pemula, karena memungkinkan kita untuk dapat terus mencoba kodingan dan melihat perubahan secara langsung yang dibuat dalam kode pada *emulator*, *simulator*, dan perangakat keras. Karena hanya dalam waktu beberapa detik, kode yang dibuah akan langsung dimuat ulang saat aplikasi berjalan tanpa perlu memulai ataupun membuild ulang. Demikian sangat bermanfaat dalam membangun UI atau menambahkan fitur tetapi juga untuk perbaikan sistem *bug*. Sejauh menyangkut kesederhanaan, *Flutter* mengklaim bahwa pemrograman dengan *flutter* sangat mudah.
4. **Kesesuaian .** Karena widget sebuah bagian dari aplikasi dan bukan platform, kita dapat mengalami lebih sedikitatau tidak adanya masalah kompibilitas tentunya pada versi *OS* yang berbeda. Ini pada gilirannya berarti lebih sedikit waktu yang akan dihabiskan untuk pengujian.
5. **Sumber Terbuka.** *Dart* dan *Flutter* keduannya bersifat *open-source* dan terlebih lagi gratis untuk digunakan dan dipakai, serta menyediakan dokumentasi yang ekstensif dan dukungan komunitas untuk membantu mengatasi masalah apa pun.

## **1. Pengenlan Python**

## **1. Sejarah Python**

## **1. Cara Kerja Python**

## **1. Kelebihan Python**

## **1.8 Kekurangan Python**

## **1.9 Manfaat Python**

# **BAB 2**

# **KEBUTUHAN FLUTTER DAN PYTHON?**

## **2.1 Hardware**

Hardware adalah perangkat fisik yang digunakan untuk mengoperasikan sebuah computer, dimana hardware sendiri meliputi semua komponen yang dapat dilihat dan dapat dirasakan seperti, monitor, keyboard, mouse, hard drive, RAM, dan lainnya. Hardware juga termasuk perangkat yang tidak dapat dilihat seperti BIOS, chipset dan lainnya,

## **2.2 Fungsi Hardware**

Hardware sendiri berfungsi sebagai pengolah data dan informasi yang dimasukkan ke dalam sebuah sistem komputer yang telah diproses. Hardware juga berfungsi sebagai penghubung computer dengan perangkat lain, seperti printer, scanner, dan jaringan. Menyediakan sumber daya yang dibutuhkan oleh sistem computer seperti daya listrik, memori, dan sebuah prosesor.

## **2.3 Ciri Bentuk Hardware**

Fisikal, hardware adalah perangkat fisik yang dapat dilihat dan dapat dirasakan seperti monitor, *CPU*, *mouse*, *keyboard* dan lain – lain. Komponen, hardware terdiri dari berbagai jenis komponen seperti chip, kabel, kartu, dan lainnya. Konektivitas, memerlukan koneksi kejaringan atau internet untuk berfungsi dengan baik. Perangkat lunak, hardware memerlukan perangkat lunak untuk dapat berfungsi seperti sistem operasi, driver, dan aplikasi. Kompatbilitas, hardware harus kompatibel dengan perangkat lain untuk dapat berfungsi dengan baik. Memberikan *Output*, sebagai hardware dapat memberikan sebuah output seperti gambar, suara, dan lain lain.

1. **DAFTAR PUSTAKA**

APA Style atau American Psychological Association adalah jenis sitasi yang diciptakan organisasi APA terutama untuk bidang psikologi dan sosial. Makanya sitasi ini seringkali ditemukan dalam karya tulis ilmiah yang ditulis oleh akademisi jurusan ilmu sosial dan psikologi. Dikutip dari lib.ui.ac.id, contoh penulisan sitasi berdasarkan APA Style.

1. Artikel Jurnal

Mellers,B. A. (2000). Choice and the relative pleasure of consequences. PsychologicalBulletin, 126, 910–924

1. Buku

Gerhardt,S. (2004). Why love matters: How affection shapes a baby’s brain. New York:Brunner-Routledge.

1. Database Online

Borman,W. C., Hanson, M. A., Oppler, S. H., Pulakos, E. D., & White, L. A. (1993).Role of early supervisory experience in supervisor performance. Journal ofApplied Psychology, 78, 443–449.Retrieved October 23, 2000, from the PsycARTICLES databas

1. Situs web

Fredrickson, B. L. (2000, March 7).Cultivating positive emotions to optimize health and well-being. Prevention & Treatment, 3, Article 0001a. Retrieved November 20, 2000, from http://journals.apa.org/prevention/volume3/pre0030001a.html

**DAFTAR PUSTAKA**

Ali, H., & Purwandi, L. (2017). Milenial Nusantara. Jakarta: PT. Gramedia Pustaka Utama.

Aziri, B. (2011). Job Satisfaction: A Literature Review. Management Research and Practice, 3(4).

Bencsik, A., Gabriella, H.-C., & Tímea, J. (2016). Y and Z Generations at Workplaces. Journal of Competitiveness, 8(3), 90–106.

Berkup, S. B. (2014). Working With Generations X And Y In Generation Z Period: Management Of Different Generations In Business Life. Mediterranean Journal of Social Sciences, 5(19).

**GLOSARIUM**

A

**Anggaran**: Suatu rencana yang disusun secara sistematis dalam bentuk angka dan dinyatakan dalam unit moneter meliputi seluruh kegiatan perusahaan untuk jangka waktu (periode) tertentu di masa yang akan datang.

B

***Budget***: Seperangkat rencana yang saling terkait satu sama lainnya yang secara kuantitatif menjelaskan proyeksi operasi perusahaan di masa depan. Rencana ini digunakan sebagai tolok ukur untuk mengukur hasil operasi *actual*, untuk alokasi dana, dan untuk rencana operasi di masa depan.

C

***Carrying******Cost***: Biaya penyimpanan

D

***Demand Forecast***: Tingkat permintaan yang diharapkan untuk produk di masa depan.

E

***Economical Order Quantity***: Jumlah pembelian ekonomis

F

***Forecast***: Proyeksi pendapatan, beban, serta pemerolehan dan penyusutan/ pembuangan aset perusahaan di masa depan.

G

***Going Concern***: Suatu Postulat yang menganggap bahwa suatu perusahaan akan terus melaksanakan operasinya sepanjang penyelesaian proyek, perjanjian, dan kegiatan yang sedang berlangsung. Perusahaan dianggap tidak berhenti, ditutup atau dilikuidasi di masa yang akan datang, perusahaan dianggap akan hidup untuk jangka waktu yang tidak terbatas (Harahap,2007)

H

I

***Indirect* Material**: Bahan Baku Tidak Langsung

**Investor**: Orang atau lembaga yang melakukan investasi dalam suatu hal dengan tujuan untuk membuat keuntungan finansial

**IT**: *Inventory Turnover*

J

K

 L

**Laporan Laba-Rugi Margin Kontribusi**: Format laporan laba rugi yang didasarkan pada pemisahan biaya menjadi komponen tetap dan komponen variabel

M

***Market Leader***: Perusahaan atau bisnis yang menguasai sebagian pasar untuk produk yang terkait.

N

**NPM**: *Net Profit Margin*

O

***Ordering Cost***: Biaya pemesanan

**OPM**: *Operating Profit Margin*

P

**Penyusunan Anggaran**: Proses pengoperasian rencana dalam bentuk unit moneter untuk kurun waktu tertentu

Q

**QR**: *Quick Ratio*

R

**Rapat Umum Pemegang Saham (RUPS)**: Organ Perseroan yang mempunyai wewenang yang tidak diberikan kepada Direksi atau Dewan Komisaris dalam batas yang ditentukan dalam Undang-undang ini dan/anggaran dasar.

S

**Standar *Usage Rate***: Standar ukuran bahan baku

T

**TDAR**: *Total Debt to Total Asset Ratio*

U

V

***Variable Budget***: Merencanakan anggaran secara sistematis dan menjelaskan secara lebih rinci tingkat perubahan biaya kegiatan perusahaan relatif terhadap biaya tidak langsung.

W

X

Y

**INDEKS**

**A**

Abraham Maslow 35 Adaptasi karakteristik 29 Adversity intelligence 6 Adversity quotient 11, 17, 61,

66, 73, 74

Affective commitment 57

Aktualisasi diri 36

Akuntan 36

Altruism 13

Amerika Serikat 23

Availability of alternative jobs

56

**B**

Bakat 64

Bakat seni 29

Bali 12

Bank 10

Basic tendencies 29 Budaya perusahaan 4

**C**

Civic virtue 13

CO2RE 66

Conscientiousness 13

Continuance commitment 57

**D**

Digital Generation 23

**E**

Edwin Locke 39

Emotional Quotient (EQ) 17

Endurance 67

Eropa 24

Etika kerja 26

**TENTANG PENULIS**

Harries Masjidanto, lahir di Kota Madiun pada tanggal 21 Agustus 1900. Pendidikan tingkat dasar hingga menengah dan atas ditempuh di Jakarta. Melanjutkan pendidikan S1 di Universitas Islam Indonesia (UII) Yogyakarta, sementara pendidikan S2 dan S3 diselesaikannya di Universitas Padjajaran (UNPAD), Bandung. Karier akademisnya diawali menjadi dosen di IISIP Lenteng Agung dan Lanlang Buana Bandung, lalu di Fakultas Ekonomi Universitas Mercu Buana (UMB), Fakultas Ekonomi UPI YAI, dan sekarang pada STIMA IMMI. Di luar aktivitasnya mengajar, penulis juga memiliki unit usaha bisnis di bidang pengembangan SDM.

A picture containing icon

Description automatically generated**Shape

Description automatically generated with low confidence**

Melihat berbagai contoh sinopsis buku mungkin terlihat mudah untuk dipraktekan langsung. Supaya memang terasa sangat mudah maka kamu perlu memahami sinopsis buku dengan mendalam. Bisa dimulai dari pengertiannya terlebih dahulu. Sinopsis buku diketahui merupakan ringkasan suatu buku yang ditulis dalam bentuk narasi. Bentuknya yang berupa ringkasan kemudian sering disalah artikan sebagai resensi. Tentunya antara sinopsis dan resensi adalah dua hal yang berbeda. Resensi juga berupa ringkasan yang condong ke arah ulasan, sehingga di dalamnya terdapat pemaparan kelebihan dan kekurangan buku. Sedangkan sinopsis murni berisi ringkasan dari isi suatu buku, sehingga tidak ada unsur kelebihan dan kekurangan di dalamnya. Sebagai ringkasan, sinopsis kemudian perlu berisi detail alur cerita dan konflik di dalam buku yang disampaikan sekilas. Selain itu, perlu diberi bumbu untuk menciptakan rasa tertarik dan penasaran bagi para pembaca.