****A picture containing icon

Description automatically generated

**Aplikasi Kelola Garansi yang Efisien dengan Golang dan PostgreSQL**

Panduan Lengkap untuk Pengembangan Backend dan Pengelolaan Database

**Roni Andarsyah**

**Megah Juliardi Sondara Wicaksana**

|  |  |
| --- | --- |
| 2023 | **PT. Penerbit Buku Pedia**  **2023** |

**Aplikasi Kelola Garansi yang Efisien dengan Golang dan PostgreSQL**

Panduan Lengkap untuk Pengembangan Backend dan Pengelolaan Database

***Penulis:***

Megah Juliardi Sondara Wicaksana

Roni Andarsyah

***ISBN***:

***Editor:***

Roni Andarsyah

***Penyunting:***

Roni Andarsyah

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

Megah Juliardi Sondara Wicaksana

***Font:***

Calibri

***Penerbit:***

PT. 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@ulbi.ac.id

Cetakan Pertama, 2023

Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk dan

dengan cara apa pun tanpa ijin tertulis dari penerbit

# PRAKATA

S

elamat datang dalam buku "Aplikasi Kelola Garansi yang Efisien dengan Golang dan PostgreSQL: Panduan Lengkap untuk Pengembangan Backend dan Pengelolaan Database". Buku ini ditujukan bagi para pengembang backend yang ingin mempelajari cara mengembangkan aplikasi kelola garansi yang efisien menggunakan bahasa pemrograman Golang dan sistem basis data PostgreSQL.

Dalam dunia bisnis saat ini, manajemen garansi menjadi hal yang sangat penting. Pelanggan mengharapkan produk yang mereka beli memiliki jaminan kualitas dan dukungan setelah pembelian. Namun, pengelolaan garansi yang tidak efisien dapat menjadi beban yang merugikan perusahaan, menghabiskan waktu dan sumber daya yang berharga.

Buku ini hadir untuk memberikan panduan langkah demi langkah dalam mengembangkan aplikasi kelola garansi yang efisien. Kami menggabungkan kekuatan bahasa pemrograman Golang yang tangguh dan sistem basis data PostgreSQL yang handal untuk menciptakan solusi yang optimal. Dalam buku ini, Anda akan mempelajari berbagai konsep dan praktik terbaik dalam pengembangan backend serta pengelolaan database untuk aplikasi kelola garansi.

Buku ini mencakup berbagai topik penting seperti autentikasi pengguna, manajemen garansi, interaksi dengan database PostgreSQL, pengembangan frontend dengan HTML dan Bootstrap, serta pengujian dan penyelesaian aplikasi. Setiap bab disusun dengan jelas dan didukung dengan contoh kode yang relevan untuk membantu Anda memahami implementasi praktis dari konsep yang diajarkan.

Kami berharap buku ini dapat menjadi sumber pengetahuan yang berharga bagi Anda dalam membangun aplikasi kelola garansi yang efisien. Dengan menggunakan panduan yang disajikan dalam buku ini, Anda akan dapat menguasai teknik-teknik pengembangan backend dan pengelolaan database yang diperlukan untuk menciptakan aplikasi kelola garansi yang handal dan responsif.

# 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

# PERSIAPAN

## Tentang Buku Ini

Buku ini, "Aplikasi Kelola Garansi yang Efisien dengan Golang dan PostgreSQL: Panduan Lengkap untuk Pengembangan *Backend* dan Pengelolaan Database", bertujuan untuk memberikan pemahaman mendalam tentang pengembangan aplikasi kelola garansi yang efisien dengan menggunakan bahasa pemrograman Golang dan sistem basis data PostgreSQL. Buku ini dirancang untuk membantu para pengembang backend dalam membangun aplikasi yang dapat mengelola proses garansi secara efisien, mulai dari pendaftaran, pemantauan, hingga penghapusan garansi.

## Tujuan dan Manfaat Buku

Tujuan utama buku ini adalah memberikan panduan langkah demi langkah yang komprehensif untuk pengembangan aplikasi kelola garansi. Buku ini mencakup aspek-aspek penting dalam pengembangan aplikasi, termasuk autentikasi pengguna, manajemen garansi, interaksi dengan database PostgreSQL, pengembangan *frontend* dengan HTML dan Bootstrap, serta pengujian dan penyelesaian aplikasi. Sedangkan untuk Manfaat dari buku ini antara lain:

1. Memahami konsep dan pentingnya kelola garansi dalam suatu sistem bisnis.
2. Menguasai bahasa pemrograman Golang dan cara mengimplementasikan fitur-fitur penting seperti autentikasi pengguna dan manajemen garansi.
3. Mempelajari cara berinteraksi dengan database PostgreSQL untuk menyimpan dan mengambil data garansi.
4. Memperoleh pengetahuan tentang pengembangan frontend dengan HTML dan Bootstrap untuk membuat tampilan antarmuka pengguna yang menarik.
5. Memahami strategi pengujian yang efektif untuk memastikan kehandalan dan kualitas aplikasi kelola garansi.
6. Mendapatkan wawasan tentang penyelesaian masalah umum dan penambahan fitur tambahan dalam aplikasi kelola garansi.

## Persyaratan dan Prasyarat

Untuk memanfaatkan buku ini secara maksimal, beberapa persyaratan dan prasyarat perlu dipenuhi. Berikut adalah beberapa yang disarankan:

1. Pengetahuan dasar tentang pemrograman dan konsep pengembangan aplikasi web.
2. Familiaritas dengan bahasa pemrograman Golang dan sistem basis data PostgreSQL.
3. Pengalaman dalam pengembangan backend menggunakan Golang.
4. Pemahaman dasar tentang HTML dan Bootstrap untuk pengembangan frontend.
5. Akses ke lingkungan pengembangan yang memungkinkan instalasi dan konfigurasi Golang, PostgreSQL, dan alat pengembangan web lainnya.

## Cara Menggunakan Buku Ini

Buku ini disusun dengan metode pembelajaran langkah demi langkah yang memudahkan pembaca dalam mengikuti konsep dan praktik pengembangan aplikasi kelola garansi. Setiap bab menyajikan materi dengan penjelasan yang terstruktur, contoh kode yang relevan, dan ilustrasi visual untuk memperjelas konsep yang dibahas.

Pembaca diharapkan untuk mengikuti urutan bab sesuai dengan daftar isi dan merujuk ke bagian-bagian yang relevan saat sedang mengembangkan aplikasi kelola garansi. Buku ini juga menyediakan contoh kode yang dapat diunduh dari situs web pendukung buku untuk membantu pembaca dalam memahami implementasi praktis dari konsep yang diajarkan.

Selain itu, buku ini mendorong pembaca untuk mengembangkan proyek aplikasi kelola garansi mereka sendiri menggunakan panduan yang disediakan. Dengan mengikuti langkah-langkah yang dijelaskan dalam buku ini, pembaca akan memperoleh pemahaman yang kuat tentang pengembangan aplikasi kelola garansi yang efisien dengan menggunakan Golang dan PostgreSQL.

# BAB 2

# PENGENALAN

## Pengantar Kelola Garansi

### Pengertian Garansi

Garansi merupakan jaminan yang diberikan oleh produsen atau penjual terhadap suatu produk yang dibeli oleh konsumen. Garansi ini memberikan perlindungan kepada konsumen dalam hal adanya kerusakan, cacat, atau masalah lain yang mungkin terjadi pada produk tersebut dalam jangka waktu tertentu setelah pembelian.

Pada dasarnya, garansi merupakan perjanjian antara produsen atau penjual dengan konsumen yang menjamin bahwa produk yang dijual bebas dari cacat atau kerusakan tertentu. Jika terjadi masalah dengan produk tersebut dalam periode garansi, produsen atau penjual akan bertanggung jawab untuk memperbaiki, mengganti, atau mengembalikan uang pembelian kepada konsumen.

Garansi memiliki beberapa elemen penting, antara lain:

* Jangka Waktu Garansi

Jangka waktu garansi menentukan berapa lama konsumen berhak memperoleh layanan garansi untuk produk yang dibeli. Jangka waktu garansi dapat bervariasi, mulai dari beberapa bulan hingga beberapa tahun, tergantung pada jenis produk dan kebijakan produsen atau penjual.

* Cakupan Garansi

Cakupan garansi mengacu pada jenis kerusakan, cacat, atau masalah yang dicakup oleh garansi. Beberapa garansi hanya mencakup kerusakan yang disebabkan oleh cacat produksi, sedangkan yang lain mungkin mencakup kerusakan yang disebabkan oleh pemakaian normal.

* Prosedur Klaim Garansi

Prosedur klaim garansi menjelaskan langkah-langkah yang harus diikuti oleh konsumen untuk memperoleh layanan garansi. Ini mungkin melibatkan pengiriman produk ke pusat layanan, menghubungi layanan pelanggan, atau mengisi formulir klaim garansi.

* Batasan dan Pengecualian Garansi

Garansi juga dapat memiliki batasan dan pengecualian tertentu. Misalnya, kerusakan yang disebabkan oleh penggunaan yang salah atau perbaikan yang tidak sah mungkin tidak dicakup oleh garansi.

### Mengapa Kelola Garansi Penting?

Garansi memiliki peran yang sangat penting dalam konteks aplikasi kelola garansi. Berikut ini adalah beberapa alasan mengapa garansi penting:

* Perlindungan Konsumen

Garansi memberikan perlindungan kepada konsumen terhadap produk yang mungkin mengalami kerusakan atau cacat. Dengan adanya garansi, konsumen memiliki jaminan bahwa mereka dapat memperoleh perbaikan atau penggantian produk jika terjadi masalah dalam periode garansi. Hal ini memberikan rasa percaya diri dan meningkatkan kepuasan konsumen.

* Kepercayaan Pelanggan

Garansi yang baik dapat membantu membangun kepercayaan pelanggan terhadap produk dan merek. Ketika pelanggan tahu bahwa produsen atau penjual memberikan jaminan atas kualitas produk, mereka cenderung lebih percaya dan yakin untuk melakukan pembelian. Garansi yang berlaku juga memberikan pesan bahwa produsen atau penjual memiliki keyakinan pada produk yang mereka tawarkan.

* Diferensiasi dari Kompetitor

Garansi yang baik dapat menjadi faktor diferensiasi yang kuat di antara pesaing. Dalam pasar yang kompetitif, memiliki garansi yang lebih baik atau lebih panjang dapat menjadi kelebihan yang menarik bagi konsumen. Hal ini dapat membantu menarik pelanggan baru, mempertahankan pelanggan yang ada, dan menciptakan keunggulan kompetitif bagi bisnis.

* Meningkatkan Reputasi dan Citra Perusahaan

Pelayanan garansi yang baik dapat membantu membangun reputasi dan citra yang positif bagi perusahaan. Ketika perusahaan memberikan dukungan yang baik selama periode garansi, ini mencerminkan komitmen mereka terhadap kepuasan pelanggan dan kualitas produk. Reputasi yang baik dalam hal garansi dapat membantu perusahaan memperoleh kepercayaan dan loyalitas pelanggan.

### Tantangan dalam Kelola Garansi

Meskipun garansi memiliki manfaat yang signifikan, pengelolaan garansi juga dapat menimbulkan beberapa tantangan yang perlu diatasi. Beberapa tantangan umum dalam kelola garansi meliputi:

* Pendaftaran Garansi yang Tidak Akurat

Salah satu tantangan yang sering terjadi adalah kesalahan dalam proses pendaftaran garansi. Pelanggan mungkin tidak mengisi informasi dengan benar atau mengabaikan langkah-langkah pendaftaran. Hal ini dapat mengakibatkan data garansi yang tidak akurat atau tidak lengkap, sehingga sulit untuk melacak dan memverifikasi klaim garansi di kemudian hari.

* Pemantauan Garansi yang Tidak Efisien

Memantau garansi secara efisien juga bisa menjadi tantangan. Terutama jika ada banyak produk dan pelanggan, mengelola data garansi, memantau masa berlaku garansi, dan mengingatkan pelanggan tentang masa garansi yang akan berakhir dapat menjadi tugas yang rumit dan memakan waktu.

* Kesulitan dalam Melacak Informasi Garansi

Melacak informasi garansi yang relevan, seperti nomor seri produk, tanggal pembelian, dan detail garansi lainnya, juga dapat menjadi tantangan. Ketika pelanggan mengajukan klaim garansi, penting untuk dapat dengan cepat mengakses informasi yang diperlukan untuk memverifikasi garansi dan memproses klaim dengan benar.

* Keterbatasan dalam Memonitor Kinerja Garansi

Mengukur kinerja garansi dan mendapatkan wawasan yang berharga tentang tren kerusakan, biaya klaim, atau tingkat kepuasan pelanggan juga dapat menjadi tantangan. Tanpa alat yang tepat untuk menganalisis data garansi, sulit untuk mengidentifikasi area yang perlu diperbaiki atau memperoleh wawasan yang dapat membantu dalam pengambilan keputusan yang lebih baik.

Mengatasi tantangan-tantangan ini memerlukan pendekatan yang efisien dan solusi yang tepat. Inilah mengapa penggunaan aplikasi kelola garansi yang efisien sangat penting untuk membantu mengelola garansi dengan lebih efektif dan efisien.

### Solusi dengan Aplikasi Kelola Garansi

Untuk mengatasi tantangan dalam pengelolaan garansi, pengembangan aplikasi kelola garansi dapat menjadi solusi yang efektif. Aplikasi ini dapat membantu mempermudah dan meningkatkan efisiensi proses pengelolaan garansi. Beberapa manfaat dari penggunaan aplikasi kelola garansi meliputi:

* Pendaftaran Garansi yang Terotomatisasi

Aplikasi kelola garansi memungkinkan pelanggan untuk dengan mudah mendaftarkan produk mereka secara online. Formulir pendaftaran yang terotomatisasi dapat memastikan pengumpulan data yang akurat dan lengkap. Hal ini membantu dalam melacak informasi garansi dengan lebih baik dan mengurangi kesalahan manusia dalam proses pendaftaran.

* Pemantauan Garansi yang Otomatis

Aplikasi kelola garansi memungkinkan pemantauan otomatis terhadap masa berlaku garansi dan memberikan notifikasi kepada pelanggan tentang masa garansi yang akan berakhir. Dengan pemantauan yang otomatis, pengelola garansi dapat dengan mudah melacak produk yang masih berada dalam periode garansi dan mengambil tindakan yang diperlukan sesuai dengan kebijakan garansi yang berlaku.

* Manajemen Klaim Garansi yang Efisien

Aplikasi kelola garansi memfasilitasi proses klaim garansi dengan mengotomatisasi langkah-langkah yang diperlukan. Pelanggan dapat mengajukan klaim garansi melalui aplikasi, yang akan mempercepat proses verifikasi dan penanganan klaim. Hal ini memungkinkan tim garansi untuk secara efisien memproses klaim, mengurangi waktu respons, dan meningkatkan kepuasan pelanggan.

* Analisis Data Garansi yang Mendalam

Aplikasi kelola garansi menyediakan alat analisis data yang kuat untuk memantau kinerja garansi, mengidentifikasi tren kerusakan, dan melacak biaya klaim. Dengan analisis yang mendalam, pengelola garansi dapat mengambil tindakan yang tepat untuk meningkatkan kualitas produk, mengoptimalkan kebijakan garansi, dan meningkatkan kepuasan pelanggan.

Penggunaan aplikasi kelola garansi dapat memberikan manfaat yang signifikan dalam efisiensi, ketepatan, dan kepuasan pelanggan. Dengan mengimplementasikan solusi ini, perusahaan dapat mengoptimalkan pengelolaan garansi, mengurangi biaya klaim, meningkatkan reputasi, dan menciptakan pengalaman pelanggan yang positif.

## Konsep dan Desain Aplikasi

### Merancang Aplikasi Kelola Garansi

Merancang aplikasi kelola garansi yang efisien membutuhkan pemikiran yang matang dan perencanaan yang cermat. Berikut adalah beberapa faktor yang perlu dipertimbangkan dalam merancang aplikasi kelola garansi:

* Identifikasi Kebutuhan Bisnis

Langkah pertama dalam merancang aplikasi kelola garansi adalah mengidentifikasi kebutuhan bisnis yang spesifik. Anda perlu memahami secara mendalam proses pengelolaan garansi yang ada dalam bisnis Anda, termasuk proses pendaftaran garansi, pengajuan klaim, pemantauan status garansi, dan analisis data garansi. Dengan memahami kebutuhan bisnis, Anda dapat merancang fitur-fitur yang sesuai dan mengoptimalkan aplikasi untuk mencapai tujuan yang diinginkan.

* Desain Antarmuka UI/UX

Desain antarmuka pengguna (UI/UX) yang baik sangat penting untuk memberikan pengalaman pengguna yang menyenangkan dan intuitif. Perancangan antarmuka pengguna yang responsif, mudah dinavigasi, dan jelas akan membantu pengguna dalam mengoperasikan aplikasi dengan lancar. Selain itu, penting juga untuk mempertimbangkan desain yang ramah pengguna, meminimalisir jumlah langkah yang diperlukan untuk menyelesaikan tugas, dan menyediakan petunjuk yang jelas bagi pengguna.

* Arsitektur Sistem yang Scalable

Merancang arsitektur sistem yang scalable (dapat berkembang seiring pertumbuhan bisnis) merupakan faktor kunci dalam merancang aplikasi kelola garansi. Anda perlu mempertimbangkan kebutuhan skalabilitas dalam hal jumlah pengguna, volume data, dan kinerja aplikasi. Arsitektur yang baik harus mampu menangani peningkatan permintaan tanpa mengorbankan kinerja dan kehandalan.

* Keamanan Data dan Privasi

Aspek keamanan data dan privasi sangat penting dalam pengelolaan garansi. Pastikan bahwa aplikasi kelola garansi yang dirancang memiliki lapisan keamanan yang kuat untuk melindungi data pelanggan dan informasi garansi dari akses yang tidak sah. Pertimbangkan penggunaan enkripsi data, otentikasi pengguna, dan pengaturan izin akses yang tepat untuk memastikan kerahasiaan dan integritas data.

* Integrasi dengan Sistem yang Ada

Aplikasi kelola garansi seringkali perlu terintegrasi dengan sistem yang sudah ada dalam bisnis, seperti sistem inventaris, sistem penjualan, atau sistem pelanggan. Pastikan bahwa aplikasi dirancang untuk mendukung integrasi dengan sistem-sistem ini dengan baik, sehingga data dapat saling terhubung dan proses bisnis dapat berjalan secara sinergis.

### Arsitektur Sistem

Arsitektur sistem adalah struktur dan organisasi komponen-komponen dalam sebuah sistem aplikasi. Dalam konteks aplikasi kelola garansi, arsitektur sistem harus dirancang dengan baik untuk memastikan kehandalan, keamanan, dan skalabilitas. Berikut adalah beberapa hal yang perlu dipertimbangkan dalam merancang arsitektur sistem untuk aplikasi kelola garansi:

* Model Client-Server

Model client-server adalah pendekatan yang umum digunakan dalam pengembangan aplikasi. Dalam konteks aplikasi kelola garansi, server berperan sebagai pusat pengelolaan data garansi dan logika bisnis. Sedangkan client (misalnya web browser) berfungsi sebagai antarmuka pengguna untuk mengakses dan berinteraksi dengan aplikasi. Model ini memungkinkan pemisahan tugas antara server dan client, sehingga memudahkan pengembangan, pemeliharaan, dan skalabilitas aplikasi.

* Backend dengan Golang

Golang (atau Go) merupakan bahasa pemrograman yang dapat digunakan sebagai backend untuk aplikasi kelola garansi. Keunggulan Golang terletak pada kinerja yang tinggi, efisiensi penggunaan sumber daya, dan kemampuan penanganan konkurensi yang baik. Dalam arsitektur sistem, backend Golang bertanggung jawab untuk mengelola logika bisnis, memproses permintaan dari client, berinteraksi dengan database, serta menyediakan API untuk komunikasi dengan frontend.

* Database PostgreSQL

PostgreSQL adalah sistem manajemen basis data relasional yang kuat dan dapat diandalkan. Dalam aplikasi kelola garansi, PostgreSQL dapat digunakan untuk menyimpan dan mengelola data garansi, informasi pelanggan, informasi produk, dan riwayat klaim. Database ini menyediakan struktur data yang fleksibel, kemampuan skalabilitas, serta dukungan untuk transaksi dan keamanan data yang penting dalam pengelolaan garansi.

* RESTful API

RESTful API (Application Programming Interface) digunakan sebagai antarmuka komunikasi antara backend dan frontend. API ini memungkinkan frontend (misalnya aplikasi web atau aplikasi seluler) untuk berinteraksi dengan backend, melakukan operasi seperti pendaftaran garansi, pengajuan klaim, dan pengambilan informasi garansi. Penggunaan standar REST memudahkan integrasi dengan berbagai platform dan pengembangan aplikasi lintas platform.

Rancangan arsitektur sistem yang baik adalah kunci untuk menciptakan aplikasi kelola garansi yang efisien, aman, dan dapat diandalkan. Dengan menggunakan Golang sebagai backend, PostgreSQL sebagai basis data, serta memanfaatkan konsep client-server dan RESTful API, aplikasi kelola garansi dapat dirancang dengan baik untuk memenuhi kebutuhan bisnis dan memberikan pengalaman pengguna yang baik.

### Komponen Aplikasi

Aplikasi manajemen garansi yang efektif terdiri dari beberapa komponen yang bekerja sama untuk menyediakan fungsionalitas manajemen garansi yang diperlukan. Berikut adalah beberapa komponen utama aplikasi manajemen garansi:

* Antarmuka Pengguna (*User Interface*)

Antarmuka pengguna atau antarmuka pengguna adalah bagian dari aplikasi yang digunakan pengguna untuk berinteraksi dengan sistem. Untuk aplikasi manajemen garansi, antarmuka pengguna dapat berupa aplikasi web yang digunakan melalui browser atau aplikasi seluler yang diinstal pada perangkat pengguna. Antarmuka ini menyediakan antarmuka yang intuitif dan mudah digunakan serta menghubungkan pengguna ke sistem backend untuk manajemen garansi, tampilan status garansi, dan fungsi lainnya.

* Backend(Logika Bisnis)

Backend adalah komponen aplikasi yang bertanggung jawab untuk mengelola logika bisnis dan menjalankan fungsi yang diperlukan untuk manajemen garansi. Backend menggunakan bahasa pemrograman Golang untuk memproses permintaan UI, berinteraksi dengan database, dan menyediakan API untuk berkomunikasi dengan UI. Komponen ini juga menerapkan aturan bisnis terkait, seperti B. validasi informasi garansi, pemrosesan klaim, dan pemantauan status garansi.

* Basis Data PostgreSQL

Database PostgreSQL digunakan sebagai gudang data dalam aplikasi manajemen garansi. PostgreSQL menawarkan struktur data yang fleksibel dan andal serta mendukung fungsi-fungsi yang diperlukan dalam manajemen garansi, seperti peristiwa B., indeks, dan keamanan data. Item ini menyimpan informasi garansi, informasi pelanggan, produk, klaim, dan riwayat lainnya.

* RESTful API

RESTful API bertindak sebagai antarmuka komunikasi antara antarmuka pengguna dan sistem backend. API ini menyediakan fungsi yang terdefinisi dengan baik seperti pendaftaran garansi, klaim, pemantauan status garansi, dan informasi garansi. Dengan menggunakan API ini, antarmuka pengguna dapat mengirimkan permintaan ke sistem backend dan menerima respons yang sesuai. Standar REST memudahkan pengembangan dan integrasi dengan aplikasi lain.

1. Desain Basis Data dengan PostgreSQL

Desain basis data yang baik menjadi fondasi yang penting dalam merancang aplikasi kelola garansi. PostgreSQL, sebagai sistem manajemen basis data yang kuat, dapat digunakan untuk menyimpan dan mengelola data garansi, informasi pelanggan, informasi produk, serta riwayat klaim. Berikut adalah beberapa langkah dalam merancang basis data dengan PostgreSQL untuk aplikasi kelola garansi:

* Identifikasi Entitas dan Relasi

Identifikasi entitas dan relasi yang akan ada dalam basis data merupakan langkah awal yang penting. Beberapa entitas yang umum dalam aplikasi kelola garansi meliputi pelanggan, produk, garansi, dan klaim. Selanjutnya, hubungan antara entitas-entitas ini perlu ditentukan, seperti hubungan one-to-one, one-to-many, atau many-to-many. Identifikasi entitas dan relasi akan membantu dalam merancang skema basis data yang efisien.

### Normalisasi Data

### Normalisasi data adalah proses untuk mengorganisasi data dalam basis data agar memenuhi aturan-aturan normalisasi. Tujuan normalisasi adalah mengurangi redundansi data, memastikan integritas data, dan meningkatkan efisiensi operasi basis data. Dalam aplikasi kelola garansi, normalisasi data dapat membantu dalam menghindari duplikasi informasi garansi, memudahkan pengelolaan data pelanggan, serta mempercepat pencarian dan manipulasi data.

### Desain Skema Basis Data

### Desain skema basis data melibatkan pemodelan struktur tabel, atribut, dan relasi antar tabel. Anda perlu mempertimbangkan dengan cermat tipe data yang sesuai untuk masing-masing atribut, penggunaan indeks untuk meningkatkan kinerja pencarian data, serta menetapkan kunci primer dan kunci asing untuk menjaga integritas referensial. Skema basis data yang baik akan memfasilitasi operasi pengelolaan garansi dengan efisien.

### Keamanan dan Akses Data

### Selain merancang struktur basis data, penting juga untuk mempertimbangkan keamanan dan akses data. Anda perlu mengatur izin akses yang tepat untuk mencegah akses yang tidak sah ke data sensitif, serta menerapkan mekanisme keamanan seperti enkripsi data atau hash password untuk melindungi kerahasiaan data pelanggan. Memiliki kebijakan keamanan yang ketat akan memberikan rasa percaya kepada pelanggan.

### BAB 3

# PENGEMBANGAN BACKEND

# DENGAN GOLANG

## Pengenalan Golang

Sebelum mempelajari pengembangan aplikasi kelola garansi dengan Golang, penting untuk memahami latar belakang dan sejarah bahasa pemrograman ini. Golang, atau juga dikenal sebagai Go, adalah bahasa pemrograman open-source yang dikembangkan di Google pada tahun 2007. Golang dirancang untuk menyediakan pemrograman yang efisien, sederhana, dan aman untuk mengembangkan aplikasi dengan skalabilitas tinggi.

Pada awalnya, Golang dikembangkan oleh tim di Google yang dipimpin oleh Robert Griesemer, Rob Pike, dan Ken Thompson. Tujuan utama dari pengembangan Golang adalah untuk mengatasi masalah yang dihadapi dalam pengembangan perangkat lunak di Google, seperti waktu kompilasi yang lambat, kompleksitas bahasa pemrograman yang tinggi, dan kesulitan dalam mengelola kode yang besar.

Rancangan awal Golang dimulai pada tahun 2007 dan perkembangannya terus berlanjut hingga dirilis secara resmi pada tahun 2009. Sejak saat itu, Golang telah mendapatkan popularitas yang signifikan di kalangan pengembang dan digunakan dalam berbagai proyek, termasuk pengembangan sistem backend, layanan web, aplikasi cloud, dan lain-lain.

Golang dirancang dengan memadukan kelebihan bahasa-bahasa pemrograman yang sudah ada, seperti kecepatan kompilasi seperti C/C++, kesederhanaan pemrograman seperti Python, dan dukungan konkurensi seperti Erlang. Hal ini menjadikan Golang sebagai bahasa yang ideal untuk pengembangan aplikasi dengan kebutuhan performa tinggi dan skalabilitas yang baik.

### Instalasi dan Konfigurasi Golang

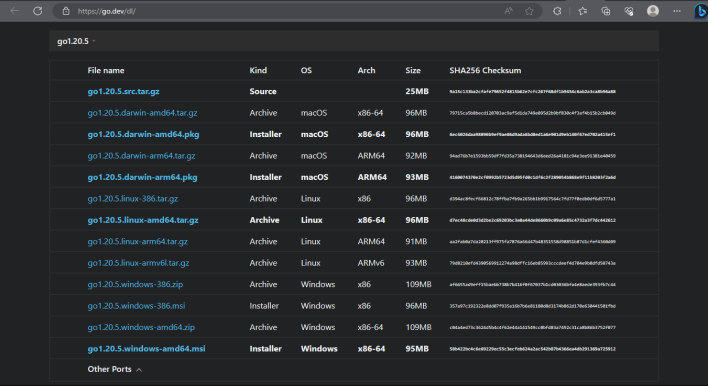
Berikut adalah langkah-langkah detail untuk menginstal Go (Golang) dari awal hingga selesai:

* Mengunduh Paket Instalasi:

- Buka situs resmi Go di [sini](https://golang.org/dl/).

- Unduh paket instalasi yang sesuai dengan sistem operasi dan arsitektur komputer Anda. Gunakan versi terbaru yang tersedia.

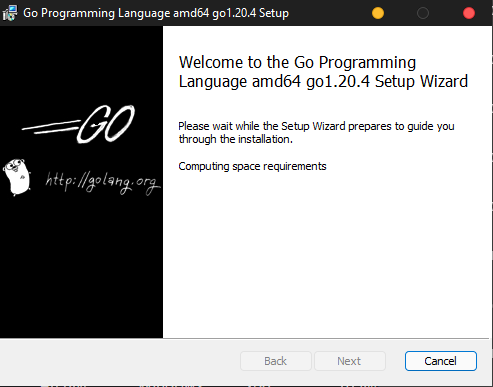
- Pilih antara instalasi berbasis MSI untuk Windows atau arsip tar.gz untuk sistem operasi Linux atau macOS dari tampilan seperti dibawah ini.



Gambar 3.1 Daftar file instalasi golang

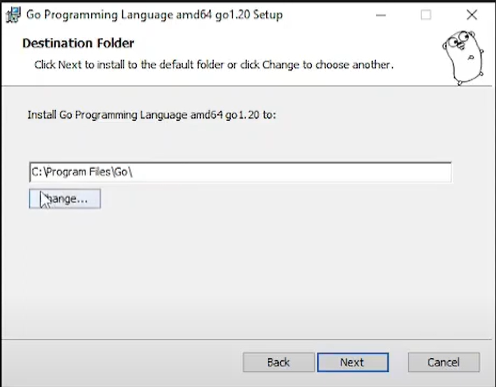
* Instalasi di Windows (Bisa juga melihat video tutorial milik Bayu Swari [disini](https://www.youtube.com/watch?v=l8FiYITnrkQ)):

- Setelah mengunduh paket MSI, buka file tersebut kemudian muncul tampilan seperti berikut

.

Gambar 3.2 Tampilan awal instalasi golang

- Ikuti petunjuk pada layar dan pilih direktori instalasi yang sesuai seperti tampilan dibawah ini.



Gambar 3.3 Tampilan pilih lokasi install

- Jika ada opsi tambahan seperti "Tambahkan Go ke PATH", centang opsi tersebut agar perintah Go dapat diakses dari baris perintah.

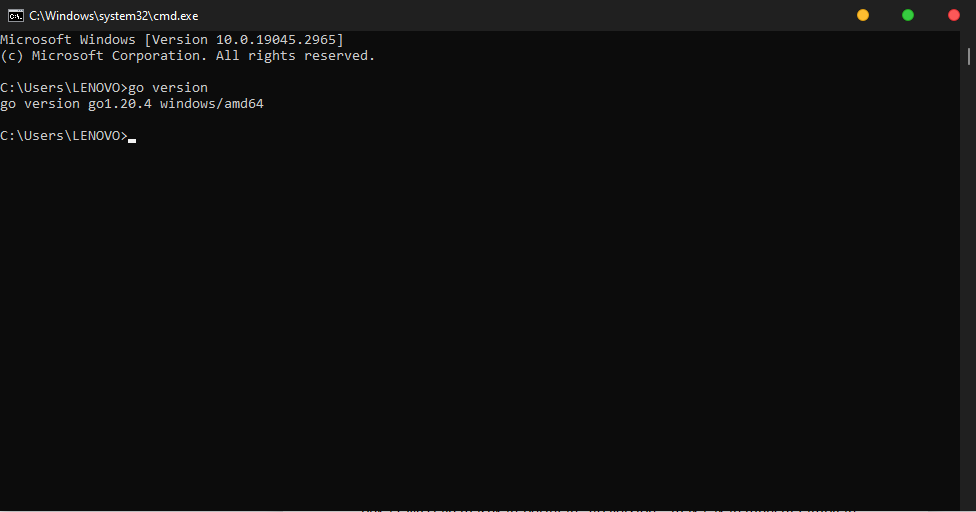
- Selesaikan proses instalasi dengan meng-klik tombol "Next" dan "Finish".

* Untuk instalasi di Linux silahkan mengikuti [video](https://www.youtube.com/watch?v=G36tXSWlUnE) ini dan di macOS mengikuti [video](https://www.youtube.com/watch?v=I2gVkuPtiuY)
* Verifikasi Instalasi:

- Buka terminal atau command prompt baru.

- Ketikkan perintah `go version` dan tekan Enter.

- Jika Anda melihat versi Go yang terinstal, berarti instalasi berhasil dan Go siap digunakan.

Gambar 3.4 Tampilan CMD saat periksa instalasi Go

* Konfigurasi Editor:

- Jika Anda menggunakan editor kode tertentu, seperti Visual Studio Code (VS Code), Sublime Text, atau GoLand, instal ekstensi Go yang sesuai untuk mendapatkan fitur penyorotan sintaksis dan bantuan pengkodean.

- Atur konfigurasi editor sesuai kebutuhan Anda, seperti menambahkan linting atau memperbarui path ke instalasi Go.

Setelah langkah-langkah ini selesai, Anda telah berhasil menginstal Go di sistem Anda dan dapat mulai menulis dan menjalankan program Go. Pastikan untuk membaca dokumentasi resmi Go dan mengeksplorasi berbagai fitur dan pustaka yang tersedia untuk memaksimalkan pengalaman pengembangan Anda.

### Struktur Proyek Golang

Dalam pengembangan aplikasi dengan Golang, penting untuk memiliki struktur proyek yang terorganisir agar mudah dipelihara dan diatur. Berikut adalah beberapa konvensi umum untuk struktur proyek Golang:

* Direktori Utama (Root Directory)

Direktori utama proyek Golang Anda biasanya merupakan direktori tingkat tertinggi di dalam workspace Anda. Direktori ini berisi file-file yang berkaitan dengan proyek, seperti file sumber kode, file konfigurasi, dan file dokumentasi.

* Direktori ‘cmd’

Direktori ‘cmd’ digunakan untuk menyimpan file-file utama aplikasi yang akan dijalankan. Biasanya, setiap subdirektori dalam ‘cmd’ berisi aplikasi yang berbeda. Misalnya, jika Anda mengembangkan aplikasi kelola garansi, Anda dapat memiliki subdirektori ‘cmd/garansi’ untuk menyimpan file utama aplikasi tersebut.

* Direktori ‘pkg’

Direktori ‘pkg’ digunakan untuk menyimpan kode yang dapat digunakan oleh aplikasi lain dalam proyek atau oleh proyek eksternal. Anda dapat membuat subdirektori di dalam ‘pkg’ untuk mengorganisir kode berdasarkan fungsionalitas atau modul tertentu.

* Direktori ‘internal’

Direktori ‘internal’ digunakan untuk menyimpan kode internal yang hanya dapat diakses oleh proyek ini dan tidak dapat diimpor oleh proyek lain. Ini memungkinkan Anda mengisolasi bagian sensitif atau tidak stabil dari kode proyek.

* Direktori ‘pkg’ dan ‘internal’ di dalam setiap subdirektori

Selain direktori ‘pkg’ dan ‘interna’l di direktori utama, Anda dapat memiliki direktori dengan nama yang sama di setiap subdirektori aplikasi. Ini memungkinkan Anda mengorganisir kode secara lebih terperinci dan membatasi akses terhadap kode yang spesifik hanya di dalam aplikasi itu sendiri.

* Direktori ‘test’

Direktori test digunakan untuk menyimpan file-file tes yang berkaitan dengan proyek. Anda dapat membuat subdirektori di dalam test untuk mengorganisir tes berdasarkan jenis atau fungsionalitas yang diuji.

### Dasar-dasar Bahasa Golang

Sebelum memulai pengembangan aplikasi dengan Golang, penting untuk memahami beberapa konsep dasar bahasa ini. Berikut adalah enam poin dasar bahasa Golang:

* Variabel dan Tipe Data

Golang menggunakan deklarasi variabel yang eksplisit dengan menentukan tipe data. Beberapa tipe data dasar yang didukung Golang antara lain: `int` (bilangan bulat), `float64` (bilangan desimal), `string` (teks), `bool` (nilai boolean), dan lain-lain. Contoh:

var age int = 25

var name string = "John Doe"

var isMarried bool = true

* Struktur Kontrol

Golang menyediakan struktur kontrol seperti `if` untuk percabangan kondisional, `for` untuk perulangan, dan `switch` untuk percabangan dengan banyak kondisi. Contoh:

if age >= 18 {

fmt.Println("Anda sudah dewasa")

}

for i := 0; i < 5; i++ {

fmt.Println(i)

}

switch day {

case "Monday":

fmt.Println("Hari Senin")

case "Tuesday":

fmt.Println("Hari Selasa")

default:

fmt.Println("Hari lainnya")

}

* Fungsi

Fungsi adalah blok kode yang dapat dipanggil dan digunakan kembali. Dalam Golang, fungsi dideklarasikan menggunakan kata kunci `func`. Contoh:

func calculateSum(a, b int) int {

return a + b

}

result := calculateSum(5, 3)

fmt.Println(result) // Output: 8

* Struktur Data

Golang mendukung struktur data seperti `struct` yang memungkinkan Anda untuk mendefinisikan tipe data kustom dengan beberapa bidang. Contoh:

type Person struct {

name string

age int

}

var p Person

p.name = "John Doe"

p.age = 25

* Slices dan Maps

Slices adalah tipe data yang fleksibel untuk menyimpan sekelompok elemen yang serupa, sementara maps digunakan untuk menyimpan pasangan nilai kunci. Contoh:

// Slices

var numbers = []int{1, 2, 3, 4, 5}

// Maps

var person = map[string]int{

"John": 25,

"Jane": 30,}

* Package dan Import

Golang menggunakan package untuk mengorganisir kode menjadi unit yang terpisah. Anda dapat mengimpor package lain untuk menggunakan kode yang ada di dalamnya. Contoh:

import "fmt"

func main() {

fmt.Println("Hello, World!")

}

Kalian bisa mempelajari dasar dasar golang lebih mendalam melalui kursus [ini (gratis bersertifikat)](https://www.sololearn.com/learning/1164)

## Autentikasi Pengguna

### Implementasi Fitur Login

### Registrasi dan Pengelolaan Pengguna

### Otorisasi dan Keamanan

## Manajemen Garansi

### Registrasi Garansi Baru

### Pencarian dan Pembaruan Garansi

### Penghapusan Garansi

## Interaksi dengan Database PostgreSQL

### Koneksi ke Database

## Operasi CRUD dengan PostgreSQL

## Optimasi Kinerja Database

# BAB 4

# PENGEMBANGAN FRONTEND

# DENGAN HTML, PHP DAN BOOTSTRAP

## Pengenalan ke HTML, PHP dan Bootstrap

### Struktur Dasar HTML dan PHP

### Menggunakan Bootstrap untuk Desain Responsif

### Membuat Tampilan Aplikasi yang Menarik

## Membangun Antarmuka Pengguna

### Desain Halaman Utama

### Halaman Login dan Registrasi

### Tampilan Tabel Garansi dan Pencarian

# BAB 5

# UJI DAN PENYEMPURNAAN

# APLIKASI

## Pengujian Aplikasi

### Strategi Pengujian

### Pengujian Unit dengan Golang

### Pengujian Integrasi dan Penerimaan

## Penyelesaian dan Penyempurnaan Aplikasi

### Optimalisasi Kinerja Aplikasi

### Penanganan Kesalahan dan Pemulihan

### Penambahan Fitur Tambahan

**BAB 6**

**PENUTUP**

**BAB 1**

**PENULISAN BUKU**

1. **FORMAT BUKU**

Penulisan menggunakan font Calibri, ukuran 11 dengan spasi 1. Margin atas dan bawah 2 cm. Margin kiri dan kanan 1,5 cm. Posisi *gutter* kiri dengan jarak 0 cm. Ukuran buku 15,5 x 23 cm. Pada bagian ini merupakan penjelasan singkat terkait pembahasan dalam satu BAB, yang dimaksudnya untuk memberikan gambaran umum kepada pembaca dan diharapkan dapat memantik motivasi pembaca untuk mempelajari sajian materi yang akan di sampaikan.

Tabel 1.1. Catatan Penting dalam Penyusunan Buku

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Petunjuk** | **Keterangan** | **Ukuran** |
| 1. | Margin atas bawah | Atas dan bawah | 2 cm |
| 2. | Margin samping | Kanan dan kiri | 1,5 cm |
| 3. | Gutter | kiri | 0 cm |
| 4. | Ukuran Buku | B5 | 15,5 x 23 Cm |
| 5. | Font | Calibri | 11 |
| 6. | Line Spacing | Jarak antar tulisan | 1 |
| 7. | Halaman redaksi | Informasi Redaksi Penerbitan setelah halaman judul menggunakan Font Book antiqua posisi melekat dari bagian bawah | 8 |
| judul yang melekat pada bagian atas dengan font Calibri 16 untuk judul dan 10 untuk sub judul | 16 dan 10 |
| 8. | Kata pengantar | Berisi data faktual penyebab buku terbit, penjelasan tata cara membaca buku dan link github dari kode program |  |

1. **TUJUAN INTRUKSIONAL DAN CAPAIAN PEMBELAJARAN**

Calibri, ukuran 11 dengan spasi 1. Secara garis besar, tujuan instruksional dalam buku ajar maupun pengajar memiliki esensi yang sama. Yaitu memberikan keterampilan, pengetahuan dan memberikan ketrampilan kepada peserta didik. Tujuan instruksional memiliki dua macam, yaitu Tujuan Instruksional Umum (TIU) dan Tujuan Instruksional Khusus (TIK).

1. ITU

TIU lebih fokus pada hasil yang harus dicapai oleh peserta didik. Berbeda dengan TIK, yang lebih menekankan pada pokok bahasan terkait dengan cabang ilmu yang lebih spesifik. Misal mengulas tentang Fisika, maka isi buku ajar mengulas banyak hal tentang ilmu Fisika.

1. TIK

TIK memiliki dua aspek yang tidak kalah penting, yaitu aspek perilaku peserta didik dan aspek pesan isi yang disampaikan. Jadi guru/dosen/pengajar secara tidak langsung berperan untuk mendidik perilaku agar berkelakuan baik dan dapat mengamalkan ilmu yang telah di dapatkan.

Komponen membuat rumusan TIK yang lengkap meliputi tiga hal, yaitu Terminal behavior, conditional of demonstration or rest dan standard of performance. Terminal behavior sering juga disebut dengan tingkah laku akhir yang diharapkan. Misalnya, siswa bisa menjadi lebih paham.

Berbeda dengan conditional of demonstration or rest atau yang disebut dengan kondisi demonstrasi. Jadi komponen pendidik diharapkan mampu memberikan demonstrasi yang mudah dipahami oleh peserta didik.

Misal, ketika guru/dosen melakukan demonstrasi, peserta didik ketika diminta untuk menjelaskan kembali, mereka bisa mendemonstrasikan ulang menggunakan cara mereka sendiri, namun inti pesan masih sama. Terakhir adalah Standard of Performance atau lebih akrab di dengar dengan standar keberhasilan. Jadi di tahap inilah pendidik/dosen/guru menekankan pada hasil yang telah dicapai.

1. **PEMBAHASAN MATERI MATERI**

Calibri, ukuran 11 dengan spasi 1, pada bagian ini berisi ulasan lengkap terkait materi pembahasan untuk tema setiap bab nya, dan dapat disajikan secara rinci melalui pembahasan sub bab. Untuk contoh bagaimana menyajikan kode program bisa dilihat pada contoh dibawah ini. Program menggunakan Bahasa pemrograman python.

def bad\_function(new\_elem, starter\_list=[]):

starter\_list.append(new\_elem)

return starter\_list

1. **RANGKUMAN MATERI**

Calibri, ukuran 11 dengan spasi 1, pada bagian ini berisi ulasan singkat terkait materi pokok pembahasan, penyajian rangkuman bisa dalam bentuk narasi atau berupa penjelasan secara numerik.

Icon

Description automatically generated

Gambar 3.1 Logo Bukupedia

1. **LATIHAN**

Calibri, ukuran 11 dengan spasi 1, di dalam buku ajar dilengkapi soal-soal latihan untuk memperdalam pemahaman siswa dan menguji kemampuan siswa dalam memahami materi yang telah disampaikan.

1. **BUKU REFERENSI**

Struktur pembasahan Memenuhi kaidah ilmiah dan estetika keilmuan yang utuh (rumusan masalah yang mengandung nilai kebaharuan, metodologi pemecahan masalah, dukungan data atau teori mutakhir yang lengkap dan jelas, kesimpulan dan daftar pusaka),

Tetapi daftar isi tidak disarankan disusun secara KAKU dengan pola IMRAD, melainkan disajikan dengan pola susunan yang lebih luwes dan dapat dipahami oleh masyarakat luas. Catatan penting mengenai buku referensi bisa dilihat pada tabel 1.1.

Tabel 1.2. Catatan Penting dalam Penyusunan Buku Referensi

|  |
| --- |
| **CATATAN PENTING YANG PERLU DICERMATI**  **PENYUSUNAN BUKU REFERENSI** |
| 1. Merupakan hasil penelitian 2. Sistematika buku referensi dan buku monograp secara umum tidak memiliki perbedaan, akan tetapi yang membedakan hanya pada perspektif dan isi pembahasan. 3. **Secara konten, buku referensi membahas persoalan melalui perspektif secara luas dalam satu bidang tertentu sesuai kompetensi penulis.** 4. Tebal buku paling sedikit 40 lembar dan berukuran standar unesco ukuran min 15.5 cm x 23 cm. 5. Tebal buku paling sedikit 40 lembar dan berukuran standar unesco ukuran min 15.5 cm x 23 cm. 6. Meskipun hasil penelitian, penyusunan struktur buku referensi disarankan tidak mengacu kepada struktur IMRAD (Introduction, Methode, Result, Discussion), tetapi disajikan dalam bentuk yang lebih luwes dan mudah di fahami oleh pembaca awam. (Kebijakan terbaru perpusnas tekait penerbitan buku hasil penelitian) |

1. **BUKU MONOGRAPH**

Struktur pembasahan Memenuhi kaidah ilmiah dan estetika keilmuan yang utuh (rumusan masalah yang mengandung nilai kebaharuan, metodologi pemecahan masalah, dukungan data atau teori mutakhir yang lengkap dan jelas, kesimpulan dan daftar pusaka),

Tetapi daftar isi tidak disarankan disusun secara KAKU dengan pola IMRAD, melainkan disajikan dengan pola susunan yang lebih luwes dan dapat dipahami oleh masyarakat luas. Catatan penting dalam penyusunan buku monograph bisa dilihat pada tabel 1.2.

Tabel 1.3. Catatan Penting dalam Penyusunan Buku Monograph

|  |
| --- |
| **CATATAN PENTING YANG PERLU DICERMATI**  **PENYUSUNAN BUKU MONOGRAPH** |
| 1. Merupakan hasil penelitian 2. Sistamatika buku referensi dan buku monograp secara umum tidak memiliki perbedaan, akan tetapi yang membedakan hanya pada perspektif dan isi pembahasan. 3. **Secara konten, buku monograp membahas persoalan dalam sudut pandang topik tertentu dalam lingkup khusus dan terbatas.** 4. Tebal buku paling sedikit 40 lembar dan berukuran standar unesco ukuran min 15.5 cm x 23 cm. 5. Tebal buku paling sedikit 40 lembar dan berukuran standar unesco ukuran min 15.5 cm x 23 cm. 6. Meskipun hasil penelitian, penyusunan struktur buku referensi disarankan tidak mengacu kepada struktur IMRAD (Introduction, Methode, Result, Discussion), tetapi disajikan dalam bentuk yang lebih luwes dan mudah di fahami oleh pembaca awam. (Kebijakan terbaru perpusnas tekait penerbitan buku hasil penelitian) |

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

C

D

E

F

G

H

I

J

K

 L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

**KREDIT GAMBAR**

Halaman ini memuat daftar kepemilikan hak cipta/perizinan penggunaan gambar/foto berikut nomor halaman keberadaan gambar/foto tersebut.

**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

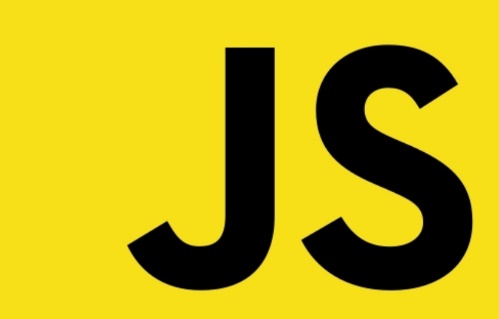
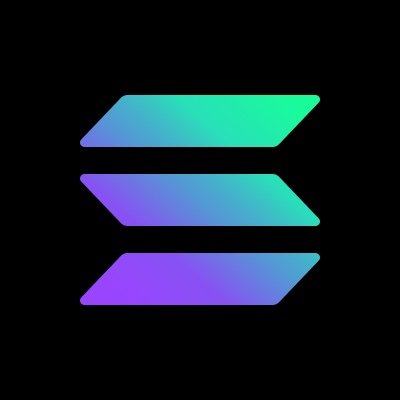
Eropa 24

Etika kerja 26

**TENTANG PENULIS**

Rolly Maulana Awangga, lahir di Kota Indramayu pada tanggal 10 November 1986. Pendidikan tingkat dasar hingga menengah ditempuh di Indramayu. Mulai merantau sejak SMA, melatih kemandiriannya di SMANDA Cirebon dengan aktif organisasi PPS Betako Merpati Putih, Pengurus OSIS dan Pendiri Dewan Keamanan Sekolah. Melanjutkan pendidikan S1 di STT Telkom, S2 di IT Telkom Bandung. Selama kuliah aktif sebagai TLH Telkom, pengurus Klub Linux Bandung, Pengurus Bandung Kota Blogger, Pendiri Saung IT dan wartawan Pikiran Rakyat. Menjadi tenaga ahli dan konsultan di aplikasi SDDKN Sekretariat Negara, Aplikasi Kementrian Hukum dan Ham, Team DevOps Pekan Olahraga Nasional, Cloud Architect Aplikasi Asesment Madrasah Kementrian Agama.

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.