

PT. Skyshi Digital Indonesia

Jl. Tampomas No. 9, Trihanggo,
Kec. Gamping, Kab. Sleman,
Daerah Istimewa Yogyakarta,
Indonesia, 55291
(0274) 4547428
hello@skyshi.com



Intern Employee Guidelines

Created by Skyshi HRD Team

Table of Contents

Table of Contents	2
Introduction	3
Intern Employee Guidelines	4
Pedoman Umum	4
Hari Kerja, Jam Kerja, dan Lembur	5
Pedoman Etika Kerja	6
Pedoman Izin Sakit	7
Pedoman Pembuatan Artikel Medium Skyshi	7
Pedoman Meeting Online	9
Tata Cara Presentasi	
Guideline Terkait	
Fundamental Skills	10
Mindset Seorang Engineer di Skyshi	10
Contoh Dokumen Bisnis Proses	13
Pedoman Tools	15
Pedoman Development	17
Pedoman Project Management	23

Introduction

Skyshi Procedure Documentation is the standard documentation for each division in managing the works to ease the bureaucratic process in carrying out the given task.

It is expected that by using this documentation the work between divisions becomes easier and more efficient.

Intern Employee Guidelines

Pedoman Umum

1. Sistem kerja fleksibel, bagi yang berada di Yogyakarta dan sekitarnya wajib hybrid 2-3 kali seminggu. Sedangkan untuk yang diluar Yogyakarta dipersilahkan jika memilih untuk full remote.
2. Hari kerja onsite tidak ditentukan, karyawan magang bebas memilih untuk onsite hari apa saja.
3. Pakaian yang digunakan adalah casual (kaos) selama tetap sopan dan rapi.
4. Komunikasi antar karyawan menggunakan aplikasi *Slack* atau *Ms Teams*.
5. Komunikasi antar karyawan lebih baik dilakukan melalui channel dalam *Slack* sehingga semua karyawan yang terkait dapat selalu update informasi melalui *Slack*.
6. Selama periode magang, karyawan menggunakan laptop pribadi.
7. Bagi yang belum menyerahkan surat keterangan magang dari kampus, bisa segera menyerahkan kepada HR.
8. HRD mengenalkan area kerja (apabila onsite) dan Lead/senior pada karyawan magang.
9. Karyawan magang dipersilahkan untuk memperkenalkan diri / menyapa terlebih dahulu kepada Lead/Senior melalui chat pribadi slack/teams agar lebih leluasa dalam bertanya dan berdiskusi kedepanya.
10. Karyawan magang diperbolehkan bertanya kepada Lead atau berdiskusi dengan rekan kerja lain terkait dengan modul yang dipelajari
11. Karyawan magang diwajibkan membuat daily report dengan update di spreadsheet yang diberikan oleh HR dan juga report di channel project atau skyshi-training setiap selesai bekerja
12. Karyawan magang diberi ruang untuk membuat artikel medium/mengisi tech talk/presentasi setiap tiga bulan sekali untuk membangun personal branding.
13. Evaluasi karyawan magang dilakukan pada bulan ke-3 minggu kedua untuk mengetahui apakah karyawan yang bersangkutan layak melanjutkan sebagai karyawan kontrak atau tidak.

Hari Kerja, Jam Kerja, dan Lembur

1. Hari dan Jam kerja

Hari Kerja	Jam Kerja	Jam Istirahat (kondisional)
Senin - Kamis	08.00 - 17.00 WIB	1 Jam 12.00 - 13.00 WIB
	09.00 - 18.00 WIB	
Jumat	08.00 - 17.00 WIB	1,5 Jam 11.30 - 13.00 WIB
	09.00 - 18.00 WIB	

Jam istirahat adalah satu jam untuk hari Senin hingga Kamis dan 1,5 jam untuk hari Jumat, mulai dan berakhir nya jam istirahat kondisional.

- Karyawan yang terlambat (datang melebihi jam 09.00) wajib memberikan alasan atau izin kepada Lead Project / PM dan HR
- Karyawan yang datang terlambat/meninggalkan pekerjaan/Pulang cepat karena alasan : melakukan hal hal yang bisa dikerjakan diluar jam kerja. Harus mengganti jam yang ditinggalkan.
- Izin meninggalkan pekerjaan yang tidak perlu mengganti jam adalah : hal hal yang tidak bisa dilakukan di luar jam kerja dan sifatnya kewajiban warga negara serta tidak bisa diwakilkan, kewajiban beribadah.
- Karyawan yang izin meninggalkan pekerjaan 3 jam atau lebih akan dipotong cuti setengah hari

2. Lembur

- Lembur harus sepengetahuan dan persetujuan dari Project Manager/System Analyst/HRD
- Melakukan pencatatan secara mandiri terkait dengan lembur yang dilakukan
- [Laporan lembur](#) diberikan kepada PM/Finance maksimal pada tgl 28 setiap bulannya

- d. Jam lembur dihitung dua jam setelah jam pulang (misal jam pulang seharusnya jam 5, maka jika yang bersangkutan masih melanjutkan pekerjaan dan pulang jam 6 belum termasuk jam lembur, namun jika yang bersangkutan masih melanjutkan pekerjaan dan pulang jam 7 maka sudah terhitung 1 jam lembur)
- e. Perhitungan untuk lembur karyawan pada jam pertama dan seterusnya adalah sebagai berikut
 - 2 x (gaji yang disepakati/173)
- f. Jam kerja maksimal dalam satu minggu adalah 54 (lima puluh empat) jam kerja (termasuk jam kerja normal dan lembur)
- g. Jam lembur maksimal dalam sehari adalah 3 (tiga) jam dan 14 (empat belas) jam dalam satu minggu
- h. Ketentuan jam kerja maksimal ini untuk membatasi karyawan menghadapi overload pekerjaan.
- i. Apabila memang dalam pelaksanaannya melebihi ketentuan tersebut, maka jam lembur akan tetap dibayarkan sesuai dengan ketentuan

Pedoman Etika Kerja

1. "Check in" melalui channel slack #absensiskysyshi2023 setiap pagi ketika akan mulai kerja
2. Update "to do" apa saja yang akan dikerjakan pada hari itu sesuai dengan task yang di assign. Update "to do" pada channel project atau channel training.
3. Update "report" task apa saja yang sudah dikerjakan setelah selesai bekerja. Maksimal report keesokan hari nya pkl 09.00 WIB. Update "report" di channel slack project atau channel training.
4. Update "progres" di file bisnis proses (sesuai dengan instruksi dari PM masing masing project).
5. Karyawan magang wajib hadir daily meet atau weekly meet yang telah ditetapkan oleh Lead Project/PM.
6. Karyawan magang diperbolehkan konsultasi atau berdiskusi dengan rekan kerja terkait dengan problem yang dihadapi dalam sebuah project

7. Karyawan magang tidak diperkenankan meminta bantuan untuk mengerjakan kesulitan atau menyelesaikan task kepada karyawan lain.
8. Task tambahan yang di assign kepada karyawan magang harus melalui sepengetahuan PM.
9. Komunikasi terkait project (pertanyaan, kesulitan, update, dll) harus dilakukan melalui channel agar semua tim mengetahui update progress project.

Pedoman Izin Sakit

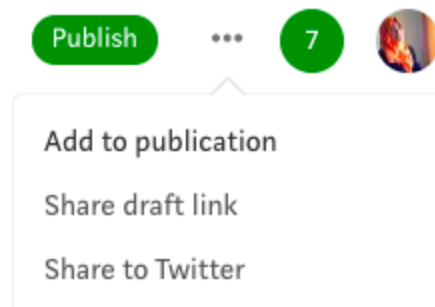
1. Karyawan magang diwajibkan untuk memberikan konfirmasi ke Lead Project/PM dan HR apabila pada hari itu berhalangan kerja karena sakit
2. Izin sakit satu hari tidak harus memakai surat keterangan dokter, tetapi jika memang sudah ada surat keterangan dokter boleh disertakan.
3. Izin sakit hari ke dua dan seterusnya wajib menggunakan surat keterangan dokter.
4. Izin sakit hari kedua dan seterusnya tanpa surat keterangan dokter akan otomatis dihitung unpaid leave.

Pedoman Pembuatan Artikel Medium Skyshi

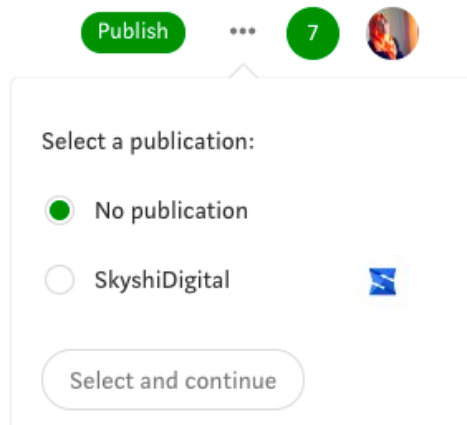
Skyshi mendorong seluruh karyawan untuk sharing ke komunitas atas ilmu yang didapat selama di Skyshi serta mendorong karyawan untuk memiliki personal branding yang bagus sesuai role nya masing masing untuk peningkatan karir karyawan. Berikut pedoman untuk membuat artikel sesuai dengan standar Skyshi :

1. Membuat akun medium menggunakan email pribadi, agar kedepanya masih bisa untuk diteruskan secara mandiri
2. Menjelaskan latar belakang persoalan
3. Materi yang dibawakan memiliki sesuatu yang berbeda
Diperbolehkan menjadikan artikel lain sebagai referensi tapi tidak diperbolehkan copy artikel lain. Masukkan konten yang orisinil, berdasarkan pemikiran dan pengalaman pribadi.
4. Menjelaskan konsep
5. Menjelaskan teknis

6. Menjelaskan secara detail
7. Hindari typo
8. Buat semenarik mungkin secara tampilan
9. Disertai Blockquote
10. Pembagian paragraf cukup jelas
11. Lama artikel (3-5 min read)
12. Setelah selesai menuliskan artikel, pastikan tidak ada typo sebelum publish
13. Konfirm kepada HR apabila siap untuk publish artikel, HR akan melakukan invitation agar karyawan magang dapat publish artikel pada medium Skyshi
14. Kemudian klik titik tiga di pojok kanan atas pada halaman write a story



15. Pilih “add to publication”



16. Pilih Skyshi kemudian “select and continue”

Pedoman Meeting secara Online

Mengingat Skyshi menerapkan sistem semi remote maka sebagian besar meeting koordinasi dilakukan secara online baik dengan tim internal, tim external maupun dengan klien. Platform yang biasa digunakan adalah Google Meet, zoom atau Ms Teams. Untuk menjaga nama baik Skyshi dan memunculkan citra profesional maka perlu memperhatikan hal hal berikut :

1. Selama melakukan panggilan pastikan berada di dalam ruangan yang cukup kondusif. Tidak terlalu bising dan mengganggu.
2. Pastikan tidak ada background yang mengganggu yang bisa memecah konsentrasi lawan bicara, atau bisa menggunakan virtual background.
3. Jika dalam meeting tidak diminta untuk open kamera, kamera bisa dimatikan. Namun jika lawan bicara open kamera, maka perlu untuk open kamera juga untuk menghormati lawan bicara.
4. Pastikan microphone berfungsi dengan baik, apabila microphone headsets tidak berfungsi dengan baik bisa diubah menggunakan microphone internal.
5. Bisa menggunakan tools meeting recording seperti tldv.io untuk menguji kualitas dari mic yang digunakan.
6. Harap testing internal dulu sebelum melakukan video call dengan klien/team untuk memastikan semua berjalan dengan baik.
7. Memastikan jaringan internet lancar sehingga suara dan gambar tertangkap dengan baik.

Fundamental Skills

Mindset Seorang Engineer di Skyshi

Fundamental First	<p>Engineer bertanggung jawab untuk bekerja smart, bukan sekedar kelihatan progressnya cepat tapi kode nya amburadul, hardcode sana sini. Good design, good structure, good reusability and good code jauh lebih penting dibanding sekedar selesai, ini akan jauh lebih long term solution.</p> <p>Engineer WAJIB memahami fundamental dari arsitektur dan modul yang digunakan. Jika belajar sebuah teknologi baru, selalu mulai dari dokumentasi dan tutorial yang diberikan dari teknologi tersebut.</p> <p>Engineer WAJIB memahami business process sebuah aplikasi. Jika mengerjakan sebuah aplikasi, selalu mulai dari business process dari aplikasi tersebut. Jika ada fitur baru atau fungsi yang dibutuhkan, maka selalu perbarui business process. Fungsi atau fitur baru yang tidak tercatat di business process akan berpotensi menjadi bug.</p>
Flexible and Responsible	<p>Skyshi memberikan fleksibilitas bagi setiap orang, namun fleksibilitas ini harus disertai dengan tanggung jawab.</p> <p>Engineer fleksibel dalam menentukan cara implementasi sebuah aplikasi, namun juga bertanggung jawab untuk mengecek sendiri pekerjaannya sesuai dengan flow yang sudah dibuat di business proses. Testing bukan tanggung jawab QA semata, tapi juga engineer.</p> <p>Engineer WAJIB “cuci piringmu sendiri”, engineer bertanggung jawab untuk tidak meninggalkan sisa pekerjaan untuk dibersihkan orang lain, jangan melimpahkan pekerjaan testing ke developer lain atau</p>

	ke QA
Do not blame the people	Engineer bertanggung jawab bekerja sebagai sebuah tim yang bekerja sama di dalam sebuah SOP. Jika ada anggota tim yang mengalami kesalahan, maka cek kembali SOP. Perbaiki SOP untuk menghindari kesalahan yang sama di project lain atau di engineer lain
Grow the people	<p>Skyshi percaya bahwa seperangkat aturan dan standar kerja yang baik akan mendorong orang - orang yang di Skyshi untuk berkembang.</p> <p>Engineer WAJIB mempunyai partner untuk melakukan code review antar anggota tim untuk saling melakukan code review pada role yang sama.</p> <p>Agar bisa self growing, setiap engineer WAJIB mempertanyakan WHY pakai ini ? Utamakan keindahan kode, reusability code, standar/protokol umum yang digunakan, terutama pada project baru yang akan dikembangkan. Jangan pernah puas. Selalu bersikap kritis pada teknologi, metode, flow, framework yang digunakan. Tidak ada kebenaran mutlak.</p>
Adding More Values	<p>Setiap orang adalah engineer di dalam dirinya, engineer itu berarti dapat mengenali problem, mengerjakan tugas dan menambahkan values lebih pada setiap hasil pekerjaannya.</p> <p>Engineer WAJIB memberikan sebuah improvisasi yang diberikan pada setiap task pekerjaannya lebih daripada hanya sekedar menjalankan tugas sesuai dengan SOP atau business process. Improvisasi ini akan menambah skills dan menambah knowledge seorang engineer. Secara long term ke depannya akan membantu growing level setiap engineer.</p>

	Engineer WAJIB mencatat di dokumentasi dan meng-komunikasi pada tim dan client tentang improvisasi yang dilakukan. Improvisasi fitur yang dilakukan dapat menjadi sebuah bug jika tidak dikomunikasikan dan tidak dicatat.
Sharing	<p>Engineer yang baik adalah engineer yang mau berbagi. Tidak ada pengalaman atau ilmu yang tidak penting untuk dibagi, biasakan berbagi pengetahuan dengan menulis ke pedoman SOP, atau ke Knowledge Management atau ke medium.com</p> <p>Engineer WAJIB membuat perancangan dan dokumentasi di tahap Inisiasi. Dokumentasi yang baik adalah cara engineer berbagi ke engineer lain tentang bagaimana sebuah aplikasi dibangun.</p>

Contoh Dokumen Bisnis Proses

Contoh Bisnis Proses

Keterangan

1. Overview IRIS Development : List summary semua sprint
2. Overview Sprint 1 : List summary untuk sprint 4 dan velocity daily
3. Sprint 4: List business process, dan task completion untuk masing - masing developer backend/frontend/mobile
4. List issue sprint 4 : Daftar issue yang muncul di sprint 4

Karakter Aplikasi yang Dibangun Skyshi

Semua aplikasi yang dikembangkan oleh tim Skyshi harus memenuhi kriteria berikut

Maintainable

Berarti aplikasi dapat di-maintain oleh tim, dan dapat dibaca oleh orang lain juga

- Aplikasi dibangun dengan business process yang jelas
- Aplikasi dibangun dengan dokumentasi yang lengkap
- Sesuai dengan guideline arsitektur yang disarankan skyshi
- Sesuai dengan guideline arsitektur yang disarankan framework

Reusable

Berarti aplikasi dibangun dengan fungsi - fungsi dan modul yang sesuai

- Kurangi hardcode, buat fungsi atau modul jika sering digunakan
- (Frontend) Tingkatkan penggunaan redux
- Notifikasi dan alert harus dibuat di modul sendiri, bukan di hardcode
- Perlu ada standar untuk log ? Pakai console.log ? atau console.tron ?

Readable

Berarti developer lain dapat membaca koding dan sistem yang dikembangkan

- Dapat dipahami nama variable atau nama fungsi nya
- Konsistensi nama variable, konsistensi input dan output API
- Konsistensi nama fungsi sesuai dengan endpoint API
- Konsistensi nama API endpoint sesuai dengan Table yang digunakan
- Konsistensi penamaan uppercase atau lowercase

Loggable dan Performance

Berarti aplikasi punya performa yang baik dan monitoring log yang lengkap

- Setiap 3rd party yang mengirimkan API harus di log

- (Frontend) Setiap perubahan navigasi harus dicatat di log
- Setiap notifikasi yang dikirimkan harus dicatat di log
- (Backend) Setiap query ke SQL harus dicatat di log, terutama execution time nya

Testable : Tahan perubahan karena punya unit testing

- Punya unit testing
- Untuk membangun confident level saat sebuah aplikasi di deploy
- Ketika sebuah fungsi ada yang diubah, maka tetap ada confident level bahwa fungsi lain tetap berjalan

Defensive Coding (optional) :

- Ada poin plus jika memperhitungkan defensive coding, misalnya ada modul yang ditambahkan diluar dari business process, untuk memperhitungkan kemungkinan extensibility aplikasi
- Contoh defensive coding : menambahkan localization, menambahkan input constraint,
- Menangani dan memperhitungkan potensial bugs yang muncul

Pedoman Tools

Employee memahami cara penggunaan macOS berikut shortcut, dan tools yang wajib ada

Referensi : Eksplorasi sendiri, Googling, medium, stackoverflow

- Employee dapat mendemonstrasikan tambah desktop baru di mac, memindahkan aplikasi ke desktop lain
- Employee dapat menginstall aplikasi diluar App Store dan memindahkannya ke folder Application sehingga aplikasi dapat dijalankan
- Employee dapat mengatur Privacy & Security untuk microphone, camera, maupun screen recording sehingga tidak mengganggu dalam berjalannya meeting online atau capture dan record screen
- Employee dapat mendemonstrasikan penggunaan terminal basic untuk berpindah folder, membuat folder baru, menghapus folder, merename folder, melakukan perintah node js seperti npm init, menjalankan aplikasi react native atau nodejs atau reactjs
- Employee dapat mendemonstrasikan pindah tab dari aplikasi terminal ke editor hanya dengan menggunakan shortcut keyboard.
- Employee dapat mendemonstrasikan penggunaan 3 jari untuk select text lalu copy paste, gunakan 4 jari untuk pindah window atau desktop
- Employee dapat mendemonstrasikan mengambil screenshot gambar atau record video dari layar, lakukan highlighting bagian yang ingin ditunjukkan, dan kirim screenshot atau record ke slack
- Employee dapat menunjukkan dan mendemonstrasikan penggunaan aplikasi berikut di laptop masing - masing

Lightshot : aplikasi screenshot, mampu membuat screenshot, highlight area

Loom : untuk record video

Tiles : untuk manage window aktif supaya dapat multi tasking antar window dalam satu desktop

Clipy : untuk mencatat riwayat clipboard agar dapat menggunakan riwayat clipboard sebelumnya

iTerm : untuk membuka terminal, mampu membuat window baru split horizontal/vertical window

Visual Studio Code : sebagai default editor

Postman : untuk membuka postman

Nvm : node version manager, lalu install node 16 dan node 18.16 LTS version, install npm dan yarn

Chrome/Firefox : untuk alternatif browser selain safari

(Tambahan untuk Backend)

Sequel Pro : untuk tools database mySQL

pgAdmin 4 : untuk tools database PostgreSQL

MongoDB Compass : untuk tools database MongoDB

(Tambahan untuk React Native)

Android Studio : dapat membuka project react native di android

XCode : dapat membuka project react native di xcode

Reactotron : dapat membuka reactotron dan menerima event dari react native

React native CLI

(Tambahan untuk ReactJS)

~~Reactotron : dapat membuka reactotron dan menerima event dari react native~~

React Devtools: Instalasi react devtools di browser dan di code

<https://www.npmjs.com/package/react-devtools>

Redux Devtools: Instalasi redux devtools di browser dan di code

<https://www.npmjs.com/package/redux-devtools>

Employee memahami pentingnya monitor kedua

Referensi : Eksplorasi sendiri, Googling, medium, stackoverflow

- Employee mampu mendemonstrasikan penggunaan monitor kedua dalam pekerjaan
- Employee mampu menjelaskan kenapa penggunaan monitor kedua membantu dalam pekerjaan

Employee wajib memahami cara kerja slack dan fiturnya

Referensi : [Using Slack](#)

- Dapat mendemonstrasikan membuat channel public dan private
- Dapat mendemonstrasikan menggunakan emoticon, dan menambahkan emoticon baru
- Dapat mendemonstrasikan membuat bold, italic, dan paste code
- Dapat mendemonstrasikan menggunakan membalas pesan, reply, share
- Dapat mendemonstrasikan membuat pin message
- Dapat mendemonstrasikan mengupload screenshot
- Dapat mendemonstrasikan meng-invite user ke channel
- Dapat mendemonstrasikan membuat automatic reply
- Dapat mendemonstrasikan membuat poll
- Memanfaatkan channel public untuk membicarakan pekerjaan, kurangi direct message

Employee memahami cara mengirimkan dan berbagi file dokumen secara online lewat Google Docs dan Spreadsheet dan Google Drive

- Dapat mendemonstrasikan membuat business process dengan Google Spreadsheet
- Dapat mendemonstrasikan membuat mapping business process dan API backend

<p>dengan Google Spreadsheet</p> <ul style="list-style-type: none"> - Dapat mendemonstrasikan membuat dokumen deployment dengan Google Docs - Dapat mendemonstrasikan sharing google docs ke public dan ke private user
<p>Engineer memahami tata cara Git branching.</p> <p>Referensi : Deployment Guideline</p> <ul style="list-style-type: none"> - Engineer memahami cara membuat git project baru - Engineer dapat clone repository git - Memahami tata cara penamaan branch - Engineer dapat men-demonstrasikan branch master, branch development, dan branch local developer, dan memahami perbedaan tujuan masing - masing branch - Memahami workflow branch di skyshi, dan tata cara merge request - Engineer dapat membuat branch baru - Engineer dapat mempush pekerjaan ke git - Engineer memahami pentingnya git commit, penulisan commit message yang baik dan memilih hanya file2 terkait dengan commit tersebut - Engineer dapat mem-pull pekerjaan dari git - Engineer dapat mensimulasikan conflict ke git - Engineer dapat melakukan solve conflict - Engineer dapat melakukan merge request dari branch lokal developer ke branch development - Engineer dapat melakukan solve conflict saat merge request

Pedoman Development

<p>Engineer dapat membaca business proses yang diberikan. Engineer dapat membuat dokumen perancangan</p> <p>Backend</p> <ul style="list-style-type: none"> - Perancangan API : Engineer dapat melakukan perancangan API endpoint yang akan digunakan. Engineer dapat melakukan perancangan output dimana sukses API, kondisi gagal API, validasi apa yang harus dilakukan - API Dummy : API dummy menggunakan Blueprint API , dan dapat dideploy ke gitlab dan hasilnya dapat diekspor ke Postman dan dapat diakses lewat Postman <p>Frontend+Mobile</p> <ul style="list-style-type: none"> - Perancangan halaman : Engineer dapat menuliskan halaman - halaman yang akan digunakan, dan redux di setiap halaman. - Perancangan redux : Engineer dapat menuliskan redux action yang akan digunakan, berisi informasi parameter input, kondisi dimana output sukses dan kondisi dimana output gagal
--

Referensi :

- Minta contoh dokumen busines process project lain
- Project pubwatch tab “Design link” (Frontend) “Response Endpoint” (Backend) ([link](#))
- [Blueprint API](#)
- [Contoh API Dummy](#)

Pahami fundamental dari core aplikasi yang digunakan. Caranya adalah dengan membaca dokumentasi, dan mengetahui fitur2 apa yang digunakan di core aplikasi tersebut, dan memahami cara kerja fitur2 tersebut.

Referensi : Googling, medium, stackoverflow**React Native/ReactJS**

Engineer memahami React native stack technology

- Engineer dapat menjelaskan cara kerja react native
- Engineer dapat menyebutkan 5 kelebihan dan kekurangan react native dibanding native android/ios
- Engineer dapat menyebutkan 3 CLI generator populer yang digunakan di react native dan kelebihan/kekurangan masing - masing
- Engineer dapat meng-implementasikan pembuatan Frontend Mobile atau Web sesuai dengan arsitektur yang ada di guideline skyshi
<https://drive.google.com/file/d/168wKWYayhq3YLH2rd1t2LUi06vDsVsjJ/view?usp=sharing>
- Engineer dapat mendemonstrasikan perbedaan require dan import,
<https://stackoverflow.com/questions/31354559/using-node-js-require-vs-es6-import-export>
- Engineer dapat menjelaskan Redux itu apa, Component itu apa, Screen itu apa
- Engineer dapat menjelaskan cara kerja Redux Saga
- Engineer dapat menjelaskan bagaimana component dan screen dapat mengakses redux
- Engineer dapat men -demonstrasikan pemanggilan redux action, dan perubahan tampilan screen, dan alert yang muncul saat terjadi perubahan state gagal, dan pindah halaman ketika berhasil mengubah state
- Engineer dapat menyebutkan beberapa package alternatif yang dapat digunakan untuk navigasi di react native/reactjs
- Engineer dapat menyebutkan beberapa package alternatif yang dapat digunakan untuk panggil API di react native/reactjs
- Engineer dapat menyebutkan beberapa package alternatif yang dapat digunakan untuk mengirimkan log di react native/reactjs
- Engineer dapat menyebutkan beberapa framework unit testing yang dapat digunakan untuk frontend

NodeJS

Engineer memahami NodeJS stack technology

- Engineer dapat menjelaskan cara kerja NodeJS
- Engineer dapat menyebutkan 5 kelebihan dan kekurangan NodeJS dibanding bahasa pemrograman lain
- Engineer dapat menyebutkan 3 tools framework populer yang digunakan di NodeJS dan kelebihan/kekurangan masing - masing
- Engineer dapat meng-implementasikan pembuatan API sesuai dengan arsitektur yang ada di guideline skyshi
<https://drive.google.com/file/d/168wKWYayhq3YLH2rd1t2LUi06vDsVsJJ/view?usp=sharing>
- Engineer dapat mendemonstrasikan perbedaan require dan import,
<https://stackoverflow.com/questions/31354559/using-node-js-require-vs-es6-import-export>
- ~~— Engineer dapat mendemonstrasikan bagaimana membuat routing di ExpressJS~~
- ~~— Engineer dapat mendemonstrasikan bagaimana membuat middleware di ExpressJS~~
- ~~— Engineer dapat mendemonstrasikan bagaimana melakukan query di ExpressJS~~
- ~~— Engineer mendemonstrasikan bagaimana menampilkan log query yang diakses, dan menampilkan kecepatan proses query~~
- ~~— Engineer mendemonstrasikan bagaimana cara melihat performance kecepatan response sebuah API endpoint~~
- ~~— Engineer dapat menyebutkan beberapa package alternatif yang dapat digunakan untuk mengirimkan log di NodeJS~~
- ~~— Engineer men-demonstrasikan unit testing yang digunakan untuk mensimulasikan API sukses dan API failed~~

QA

- Engineer dapat menjelaskan cara kerja Cypress
- Engineer dapat menyebutkan 5 kelebihan dan kekurangan Cypress dibanding automation testing lain
- Engineer dapat mendemonstrasikan bagaimana membuat automation testing untuk sebuah flow register hingga login di sebuah web

Engineer memahami pentingnya melakukan automation testing. Engineer mampu membuat unit testing

Referensi : Eksplorasi sendiri, Googling, medium, stackoverflow

Backend

- Engineer mampu mendemonstrasikan automation testing untuk pemanggilan API endpoint yang menampilkan kondisi sukses dan kondisi gagal (API gagal karena gagal validasi request)
- Engineer mampu mendemonstrasikan automation testing untuk sebuah flow business process

- Engineer mampu mendemonstrasikan menjalankan automation testing pada saat deployment di gitlab

Frontend+Mobile

- Referensi contoh kode <https://gitlab.skyshi.io/jaknot/jaknot-mobile/tree/development/Tests>
- Unit testing Redux: Engineer mampu mendemonstrasikan automation testing untuk pemanggilan redux action, dan mendemonstrasikan kondisi sukses dan kondisi gagal
- Unit testing Component: Engineer mampu mendemonstrasikan automation testing untuk pemanggilan component, dan mendemonstrasikan kondisi sukses dan kondisi gagal
- Unit testing Services: Engineer mampu mendemonstrasikan automation testing untuk pengujian API services, dan mendemonstrasikan kondisi sukses dan kondisi gagal
- Engineer mampu mendemonstrasikan menjalankan automation testing pada saat deployment di gitlab

QA

- Engineer mampu mendemonstrasikan automation testing untuk sebuah business process, dan mendemonstrasikan kondisi sukses dan gagal
- Engineer mampu mendemonstrasikan menjalankan automation testing pada saat deployment di gitlab

Engineer dapat menuliskan coding sesuai dengan style guideline

Backend :

- NodeJS style guide <https://github.com/felixge/node-style-guide>

Frontend + Mobile :

-

Engineer dapat melakukan deployment otomatis ke deployment server, gunakan pipeline gitlab

Referensi : [Deployment Guideline](#)

Backend + frontend + Mobile + QA

- Engineer men-demonstrasikan deployment otomatis
Backend : deploy API otomatis
Frontend : deploy Web otomatis
Mobile : build APK otomatis
QA : run Automation testing
- Engineer men-demonstrasikan bahwa response API atau tampilan web atau APK atau automation testing akan berubah ketika dilakukan perubahan commit baru ke git

(Frontend+Mobile+Backend only)

Engineer memahami tata cara pembuatan monitoring tools. Setiap pendaftaran 3rd party seperti notifikasi email, monitoring tools newrelic dan sentry gunakan email

developer+namaproject@skyshi.com

Referensi : Eksplorasi sendiri, silakan request info login developer@skyshi.com

- Developer dapat membuka dan membaca email developer@skyshi.com
- Developer dapat mendaftar layanan sentry dan menambah project baru menggunakan email developer+namaproject@skyshi.com
- Developer dapat mendaftar layanan loggly dan menambah project baru menggunakan email developer+namaproject@skyshi.com
- Developer dapat mendaftar layanan newrelic dan menambah project baru menggunakan email developer+namaproject@skyshi.com
- Developer dapat mendaftar layanan notifikasi email seperti postmarkapp menggunakan email developer+namaproject@skyshi.com

Engineer memahami pentingnya code review

Referensi : [Code Review Guideline](#)

- Engineer dapat menjelaskan kenapa perlu code review
- Engineer dapat menjelaskan poin - poin penting yang harus diperiksa pada saat code review
- Engineer dapat melakukan proses merge request dan merge approval
- Engineer dapat menuliskan saran dan komentar pada halaman merge request

Engineer memahami pentingnya log dan validasi input API.

Demonstrasikan unit testing, validasi input API dan log. Gunakan perancangan API dummy yang dibuat di tahap sebelumnya.

Referensi : [Firebase \(Mobile\)](#) [Sentry](#) dan [Loggly](#) dan [Cypress Documentation](#)

Backend

- Engineer dapat menjelaskan perbedaan kapan menggunakan Sentry dan Loggly
- Engineer dapat men-demonstrasikan validasi gagal ketika sebuah input dikirim ke API endpoint
- Engineer dapat men-demonstrasikan log dikirim ketika API endpoint dipanggil oleh user berisi informasi log request dan response dan kecepatan response
- Engineer dapat men-demonstrasikan log dikirim ketika ada query ke SQL berisi query dan response
- Engineer dapat men-demonstrasikan log dikirim ketika 3rd party API sukses/gagal dipanggil
- Engineer dapat men-demonstrasikan log dikirim ketika aplikasi crash

Frontend + Mobile

- Engineer dapat men-demonstrasikan validasi gagal ketika sebuah input dikirim ke mobile apps
- Engineer dapat men-demonstrasikan log dikirim ketika navigasi berubah oleh user

- Engineer dapat men-demonstrasikan log dikirim ketika 3rd party API sukses/gagal dipanggil
- Engineer dapat men-demonstrasikan log dikirim ketika aplikasi crash

QA

- QA melakukan process automation testing end-to-end
- QA dapat mendemonstrasikan automation testing untuk validasi yang gagal dan validasi yang berhasil
- QA dapat men-demonstrasikan automation testing untuk mengubah navigasi
- QA dapat men-demonstrasikan automation testing untuk memeriksa halaman tersebut sukses diload atau tidak
- QA memahami pembuatan random email menggunakan primary+blabla@skyshi.com

Engineer memahami cara menangani error dan exception handling di code, baik karena expected error (misal gagal validasi constraint, gagal validasi JWT), atau unexpected error (misal karena gateway timeout, atau database timeout)

Frontend+Mobile

- Engineer dapat mendemonstrasikan response aplikasi ketika internet tidak terhubung
- Engineer dapat mendemonstrasikan response aplikasi ketika server mati
- Engineer dapat mendemonstrasikan log dikirim ke sentry ketika ada expected error
- Engineer dapat mendemonstrasikan log dikirim ke sentry ketika ada unexpected error

Backend

- Engineer dapat mendemonstrasikan penggunaan try catch pada sync code
- Engineer dapat mendemonstrasikan penggunaan try catch pada async code
- Engineer dapat menunjukkan API response error ketika terjadi expected error (gagal validasi constraint, ataupun JWT)
- Engineer dapat menunjukkan API response error ketika terjadi unexpected error (database timeout atau gateway timeout)
- Engineer dapat mendemonstrasikan penggunaan try catch yang benar pada ruang lingkup yang tepat

Baca :

- <https://stackoverflow.com/questions/7310521/node-js-best-practice-exception-handling>
- <https://medium.com/@Abazhenov/using-async-await-in-express-with-node-8-b8af872c0016>
- Engineer dapat mendemonstrasikan log dikirim ke sentry ketika ada expected error
- Engineer dapat mendemonstrasikan log dikirim ke sentry ketika ada unexpected error

Pedoman Project Management

Engineer memahami project management di skyshi

Referensi : [Project Management Guideline](#)

- Engineer dapat menyebutkan 4 tahapan project di Skyshi
- Engineer dapat menjelaskan output dokumen dari setiap tahap
- Engineer dapat menjelaskan siapa yang terlibat dan jobdesc di setiap tahapan
- Engineer dapat membaca dan membuat business process
- Engineer dapat melakukan breakdown task dari business process
- Engineer dapat melakukan ekspor task breakdown ke google spreadsheet task
- Engineer dapat melakukan dan mendorong daily standup setiap hari dengan tim

Engineer memahami pentingnya penggunaan zoho dan melakukan to do dan report dalam kolaborasi team, dan sebagai transparansi pekerjaan ke klien

Referensi : [Project Management Guideline](#)

- Engineer dapat memberikan todo list pekerjaan dan estimasi task yang akan “done” setiap pagi berdasarkan task yang sudah disiapkan di tahap inisiasi
- Engineer dapat memberikan laporan progress, jumlah task yang “done” setiap hari dan mengupdate status di ZOHO atau project management (spreadsheet yang dibuat PM) yang digunakan.
- Engineer dapat menjawab berapa persen, berapa jumlah task yang “done” setiap hari dan persentase dari sebuah project, serta berapa jumlah task perhari dan persentase untuk backend, frontend, ataupun mobile

Engineer memahami cara berkomunikasi tentang masalah project ke klien

Referensi : [Project Management Guideline](#) (Perlu tata cara komunikasi di dokumen terpisah?)

- Engineer memahami apa yang harus dilakukan jika ada perubahan fitur. Apakah ini bug ? atau apakah ini fitur ? Semua tergantung pada business process, Sebuah fitur baru mempunyai resiko tinggi menjadi bug jika tidak ada di business process.
- Engineer dapat menjelaskan tata cara berkomunikasi dengan klien, jika ada problem maka diskusikan di internal dulu
- Engineer dapat menjelaskan tata cara berkomunikasi dengan engineer lain, bisa menjelaskan apa yang sudah dilakukan, dan apa yang menjadi masalah, dan penelusuran apa yang sudah dilakukan
- Engineer memahami bahwa setiap kali pertanyaan yang disampaikan ke klien, harus dicek dulu ke business process ada atau tidak, jika tidak ada maka baru ditanyakan. Sebaliknya juga, setiap kali ada usulan fitur baru, apakah sudah ada di business process, jika tidak ada maka sampaikan ke klien juga.

Engineer memahami tata cara perbaikan bug

Referensi : [Project Management Guideline](#) dan [Backlog Knowledge Management System](#)

- Engineer dapat mensimulasikan mendapat laporan bug dari client,
- Engineer dapat menduplikasi laporan error/bug
- Engineer melakukan pencarian bug ini di knowledge management system
- Kumpulkan fakta sebanyak2nya, kumpulkan data sebanyak2nya, kumpulkan log
- Buat percobaan sebanyak2nya, cari pola nya, kapan bug tersebut terjadi
- Buat hipotesis, apa yang menjadi perkiraan penyebab
- Lakukan eksperimen kembali berdasarkan hipotesis
- Buat kesimpulan dan rekomendasi
- Lakukan perbaikan
- Simpan hasil rekomendasi pada knowledge management system

Engineer mempunyai sebuah masalah atau kebingungan dalam mengerjakan sebuah aplikasi atau task

Referensi : [Project Management Guideline](#) dan [Backlog Knowledge Management System](#)

- Engineer dapat mensimulasikan melakukan pencarian masalah di knowledge management system
- Kumpulkan fakta sebanyak - banyaknya, kumpulkan data sebanyak-banyaknya, kumpulkan log
- Buat hipotesis, beberapa solusi yang dapat digunakan
- Lakukan eksperimen kembali berdasarkan hipotesis
- Buat kesimpulan dan rekomendasi
- Simpan hasil rekomendasi pada knowledge