

Perancangan aplikasi mobile *to-do list* sebagai platform delegasi tugas secara *remote* pada masa *pandemic* di pt. pupuk kujang cikampek

Robby Akbar^{#1}, Taufik Ridwan^{#2}

[#]Universitas Pendidikan Indonesia, Kampus Purwakarta

¹robbyakbar@upi.edu, ²taufikridwan@upi.edu

Abstract - In the current pandemic, many changes have occurred in human life. One of them is that companies do work from home virtue to some of their employees. In an effort to prevent the massive spread of the Covid-19 virus and minimize the addition of cases to industrial clusters. However, there are some obstacles that occur during the transition to working from home, such as assigning tasks that are usually done directly by managers to employees being less effective if done via messages on messaging platforms. So that it causes miss information, and has an impact on doing tasks that are not as ordered / expected. The purpose of this research is to design a Mobile To-do List application at PT. Pupuk Kujang Cikampek. The method used is the Software Development Life Cycle (SDLC), which includes the planning, analysis, design, implementation, testing, and maintenance stages. This research has reached the implementation stage of the design into an application, which has several features, namely facilitating and assisting a manager in delegating tasks to employees, such as creating projects that contain tasks in one unit and being able to directly determine who employees will do them. In addition, it can also monitor employee work progress, and can discuss / provide comments on tasks that have been done. As well as several other supporting features that make this application even more efficient to use during the current pandemic. It is hoped that the To-do List application can help managers and employees coordinate to carry out their work tasks in the company, according to the needs needed in the era of the digitalized 4.0 industrial revolution.

Keywords – Delegation Task, Pandemic, Remote, To-Do List.

Abstrak – Di masa *pandemic* saat ini, banyak perubahan yang terjadi dalam kehidupan manusia. Salah satunya adalah perusahaan-perusahaan melakukan kebijakan *work from home* kepada beberapa pegawainya. Sebagai upaya mencegah penyebaran virus Covid-19 yang masif, dan meminimalisasi penambahan kasus pada *klaster* industri. Namun, terdapat beberapa kendala yang terjadi dalam masa transisi untuk bekerja dari rumah, seperti pemberian tugas yang biasanya dilakukan secara langsung oleh manajer kepada karyawannya menjadi kurang efektif jika dilakukan melalui pesan di platform *messaging*. Sehingga menyebabkan *miss* informasi, dan berdampak terhadap pengerjaan tugas yang tidak sesuai dengan yang diperintahkan/diharapkan. Tujuan dari penelitian ini adalah untuk merancang aplikasi Mobile *To-do List* di PT. Pupuk Kujang Cikampek. Metode yang digunakan adalah *Software Development Life Cycle* (SDLC), yang meliputi tahapan rencana, analisis, desain, implementasi, uji coba, dan pengelolaan. Penelitian ini sudah sampai tahap implementasi rancangan menjadi sebuah aplikasi, yang terdapat beberapa fitur yaitu memudahkan dan membantu seorang manajer dalam mendelegasi tugas kepada karyawannya, seperti membuat proyek yang berisikan tugas-tugas dalam satu unit dan dapat langsung menentukan siapa karyawan yang akan mengerjakannya. Selain itu juga dapat memantau *progress* pekerjaan karyawan, dan dapat mendiskusikan/memberikan komentar terhadap tugas yang sudah dikerjakan. Serta beberapa fitur pendukung lainnya yang membuat aplikasi ini semakin efisien untuk digunakan pada masa *pandemic* saat ini. Diharapkan, aplikasi *To-do List* dapat membantu manajer dan karyawan dalam berkoordinasi untuk menjalankan tugas-tugas pekerjaannya di perusahaan, sesuai dengan kebutuhan yang diperlukan di era revolusi industri 4.0 yang serba digitalisasi.

Kata kunci – Catatan, Delegasi Tugas, Jarak Jauh, Pandemi.

I. PENDAHULUAN

Pandemi Covid-19 yang terjadi di penghujung tahun 2019 tepatnya di Wuhan China membuat banyak perubahan dalam kehidupan manusia saat ini. Pada 2 Maret 2020, Presiden Joko 'Jokowi' Widodo mengumumkan kasus pertama (dan kedua) positif terinfeksi virus korona baru, atau terkena penyakit COVID-19, di Indonesia [1]. Dalam pidatonya

menginstruksikan untuk masyarakat Indonesia mengurangi kegiatan di luar rumah yang tidak penting. Termasuk menerapkan sistem kerja *Work From Home* dan sekolah serta kuliah secara Online [2]. *Work from home* telah mengubah sistem kerja yaitu melakukan pelayanan kepada *public* melalui daring (*online*) [3].

Dalam prakteknya pelayanan *online* merupakan bentuk implementasi dari protokol pencegahan Covid-19 yang diberlakukan oleh pemerintah. Namun, tidak semua

perusahaan memiliki kesiapan untuk menghadapi ini. Salah satunya dalam melakukan komunikasi internal yang dilakukan antara manajer dengan pegawai (komunikasi vertikal), seperti pemberian tugas, arahan, pengawasan, dan pemberian apresiasi dari atasan kepada bawahan. Proses komunikasi ke atas berupa laporan secara tulisan maupun lisan dari bawahan kepada atasan [4].

Komunikasi vertikal yang dilakukan secara daring dapat melalui beberapa cara seperti surat elektronik, dan aplikasi media sosial. Penggunaan aplikasi tersebut sangat membantu untuk komunikasi jarak jauh, namun yang menjadi pertimbangan adalah ketidaksesuaian tujuan aplikasi tersebut. Surat elektronik diperuntukkan sebagai platform untuk mengirimkan pesan secara formal, dan *chat* melalui media sosial biasanya berupa teks singkat untuk memberikan/menanyakan kabar.

Platform tersebut dapat digunakan dalam melakukan komunikasi vertikal antara manajer dan karyawan, tetapi tidak efektif untuk dilakukan dalam skala besar. Sehingga akan timbulnya masalah seperti terjadinya *miss* informasi, yang diakibatkan karena pesan pemberian tugas yang bertumpuk dengan pesan yang lainnya. Selain itu tidak dapat memfasilitasi manajer memberikan komando kepada banyak karyawan (tim) sekaligus, yang bisa berdampak terhadap pengerjaan tugas tidak sesuai dengan yang diperintahkan/diharapkan.

Berdasarkan tinjauan pustaka yang peneliti lakukan, terdapat beberapa aplikasi lain yang bisa digunakan untuk mendelegasi tugas secara *remote* seperti Asana, Trello, dan Taiga.io. Yang memiliki kelebihan dan kekurangannya masing-masing. Jika dibandingkan dengan aplikasi *To-do List* yang akan dikembangkan, tentunya memiliki perbedaan seperti, fitur-fitur yang dibuat sesuai dengan kebutuhan yang ada. Adanya pelaporan progres berupa persentase dari setiap *task* yang diberikan, dan sistem yang terintegrasi dengan sistem kepegawaian milik perusahaan melalui *single sign on*. Sehingga tidak perlu khawatir terhadap kerahasiaan data privasi perusahaan. Kelebihan lainnya dari aplikasi ini adalah membatasi tugas prioritas dalam satu hari yang tidak bisa lebih dari satu tugas, sehingga membuat karyawan lebih fokus terhadap satu tugas, dan hasil yang didapatkan maksimal. Adanya fitur laporan berupa statistik untuk mengetahui kinerja yang telah dilakukan karyawan dari *project* atau *task*.

Dengan mempertimbangkan fitur yang dibutuhkan oleh perusahaan terhadap platform delegasi tugas, maka peneliti memilih untuk merancang dan mengembangkan aplikasi *To-do List* dengan desain sendiri (*self-design*). Untuk mengetahui fitur-fitur apa saja yang memang dibutuhkan perusahaan untuk memberikan tugas kepada setiap karyawannya. Diharapkan aplikasi ini dapat membantu manajer dan karyawan dalam berkoordinasi untuk menjalankan tugas-tugas pekerjaannya di perusahaan, sesuai dengan kebutuhan yang diperlukan di era revolusi industri 4.0 yang serba digitalisasi.

II. METODE PENELITIAN

Pada penelitian ini menggunakan metode pengembangan sistem *Software Development Life Cycle* (SDLC). Menurut Susanto dan Andriana (2016:42) SDLC adalah proses pembuatan dan perubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem-sistem. Konsep ini umumnya merujuk pada sistem komputer atau informasi. SDLC juga merupakan metodologi umum dalam pengembangan sistem yang menandai kemajuan usaha analisis dan desain [5]. Tahapan umum dari SDLC meliputi rencana, analisis, desain, implementasi, uji coba, dan pengelolaan. Metode ini memiliki turunan atau sebagai dasar dari berbagai model/metodologi pengembangan perangkat lunak.

III. HASIL DAN PEMBAHASAN

A. PERENCANAAN SISTEM

Untuk menjawab permasalahan yang ada di perusahaan dalam memberikan tugas ke karyawannya yang bekerja secara *remote*, peneliti melakukan wawancara kepada salah satu manajer untuk menentukan fitur-fitur yang akan dikembangkan pada aplikasi *To-do List*. Hasil yang diperoleh dapat disimpulkan pada tabel 1.

Tabel 1. Fitur pada Aplikasi *To-do List*

ID	Menu	Deskripsi
CP	<i>Create Project</i>	Manajer dapat membuat proyek yang berisikan tugas-tugas dalam satu unit, lalu menentukan siapa karyawan yang akan mengerjakannya.
CT	<i>Create Task</i>	Manajer maupun karyawan dapat membuat tugas harian yang akan dikerjakan.
PT	<i>Project & Task</i>	Karyawan dapat melihat daftar tugas yang telah diberikan oleh manajer, maupun tugas individu yang telah dibuat sendiri.
RP	<i>Report Progress</i>	Karyawan dapat melaporkan perkembangan tugas yang sedang dikerjakan.
MP	<i>Monitor Progress</i>	Manajer dapat memantau <i>progress</i> pekerjaan karyawan, melalui laporan <i>progress</i> yang telah dibuat oleh karyawan.
DP	<i>Discuss Progress</i>	Manajer dan karyawan dapat mendiskusikan atau memberikan komentar terhadap progres yang telah dilakukan.

Serta beberapa fitur pendukung lainnya yang membuat aplikasi ini semakin efisien untuk membantu proses pemberian, pelaporan, dan pemantauan tugas.

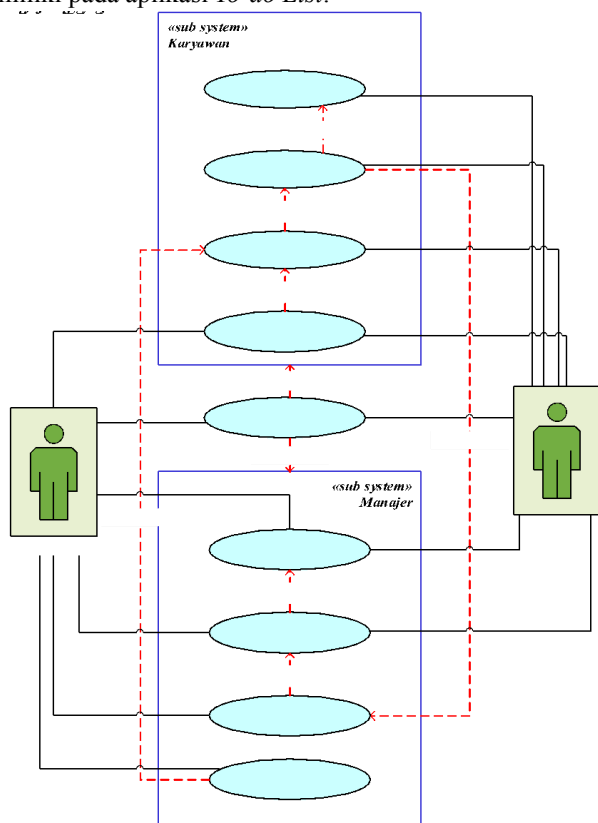
B. ANALISIS KEBUTUHAN SISTEM

Berdasarkan rencana yang telah disusun, selanjutnya melakukan analisis kebutuhan. Analisis kebutuhan menurut Majid *et al.* "An analysis of the requirements is the collection of requirements needed to develop a complete application" [6]. Setidaknya bertujuan untuk

mengetahui bagaimana cara kerja aplikasi tersebut, alur proses data dari aplikasi, serta bentuk/struktur dari data aplikasi. Berturut-turut disajikan dalam bentuk (1) UML berupa *Use Case Diagram*, (2) *Data Flow Diagram*, dan (3) *Database Diagram*.

1. *Use Case Diagram*

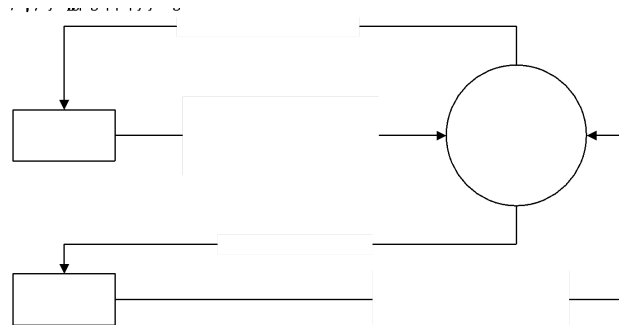
Gambar 1 merupakan *use case diagram* dari aplikasi *To-do List*. Terdapat 2 aktor yang berperan dari sistem ini yaitu manajer, dan karyawan. Otorisasi dari manajer yang dapat membuat/memberikan tugas kepada karyawan, dapat memantau progres pekerjaan, dan mendiskusikan progres yang telah dibuat karyawan. Sedangkan otorisasi karyawan dapat membuat *task* individu, mengambil/mengerjakan tugas, melaporkan progres, dan membuat laporan dari pekerjaan yang telah dilakukan. Untuk dapat melakukan hal tersebut, manajer dan karyawan terlebih dahulu harus *login* menggunakan akun masing-masing untuk menentukan otoritas yang mereka miliki pada aplikasi *To-do List*.



Gambar 1. Use Case Diagram Aplikasi To-do List.

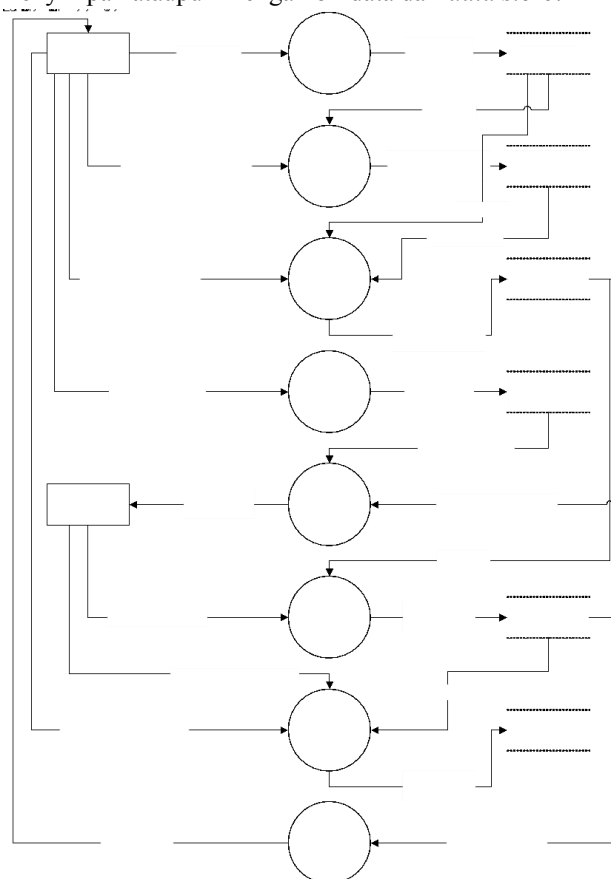
2. *Data Flow Diagram*

DFD berfungsi untuk menggambarkan aliran data yang terjadi pada sebuah sistem. Dengan adanya DFD dapat menjadi gambaran bagi pengembang untuk menentukan skema *database* yang akan dibuat. Dalam merancang DFD haruslah sesuai dengan kebutuhan informasi yang akan diperoleh pengguna. Gambar 2 merupakan DFD level 0 dari aplikasi *To-do List*. Terdapat 2 entitas yang berpengaruh secara langsung terhadap sistem, yaitu manajer, dan karyawan.



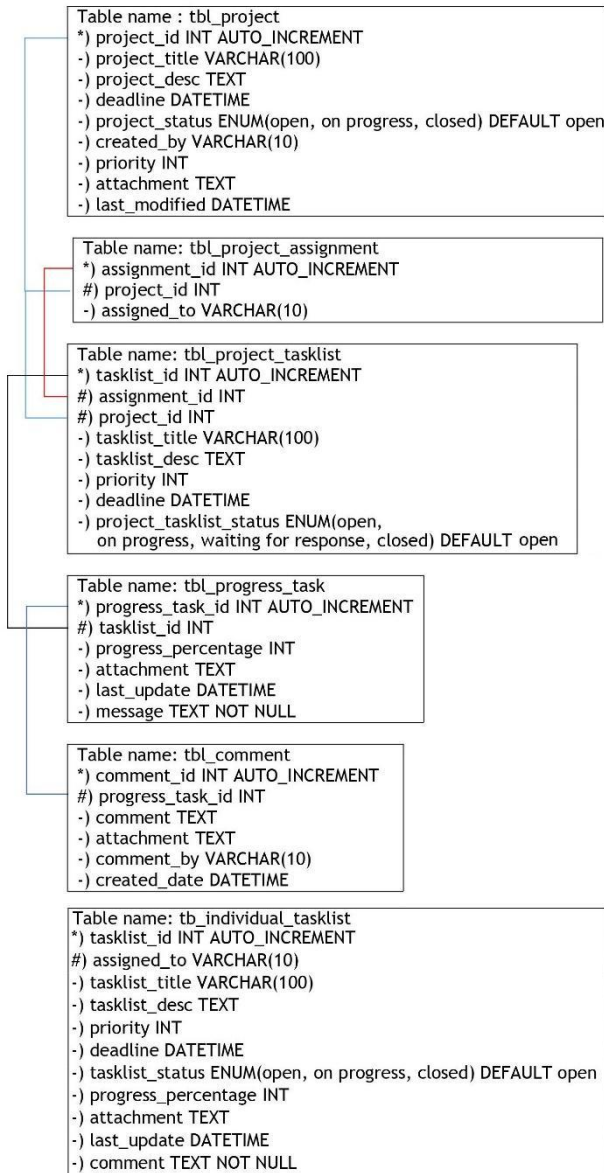
Gambar 2. DFD level 0 Aplikasi To-do List.

Dari DFD level 0 dapat dilakukan *breakdown* ke DFD level 1 untuk mengetahui lebih detail tujuan data disimpan dan diperoleh. Selain itu DFD 1 juga untuk mengetahui proses yang terjadi ketika data-data dialirkan dari sistem ke entitas maupun sebaliknya. Jumlah proses yang terdapat pada DFD level 1 setidaknya tidak kurang dari jumlah aliran data yang ada pada DFD level 0. Pada gambar 3 merupakan DFD level 1, terdapat 6 tabel untuk menyimpan data-data, dan 9 proses yang terjadi untuk menyimpan ataupun mengambil data dari *data store*.



Gambar 3. DFD level 1 Aplikasi To-do List.

3. *Database Diagram*

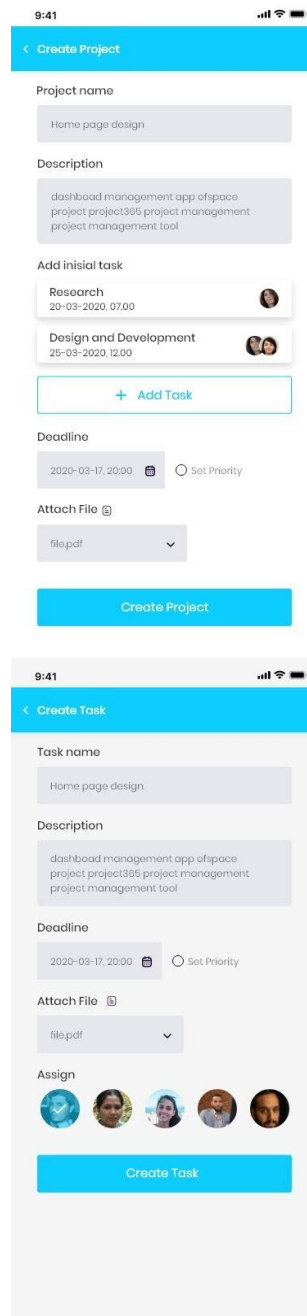


Gambar 4. Database Diagram Aplikasi To-do List.

Gambar 4 merupakan skema/design database yang menjelaskan lebih lanjut dari data store yang ada pada DFD level 1, dan pada Database Diagram dibuat lebih detail data-data yang terdapat dari sebuah tabel. Nama kolom, panjang data, tipe data, hingga hubungan data dari suatu tabel ke tabel lainnya ditentukan di Database Diagram.

C. DESAIN TAMPILAN APLIKASI

Pada tahap ini menghasilkan beberapa tampilan mockup awal dari aplikasi To-do List, seperti pada gambar-gambar berikut:



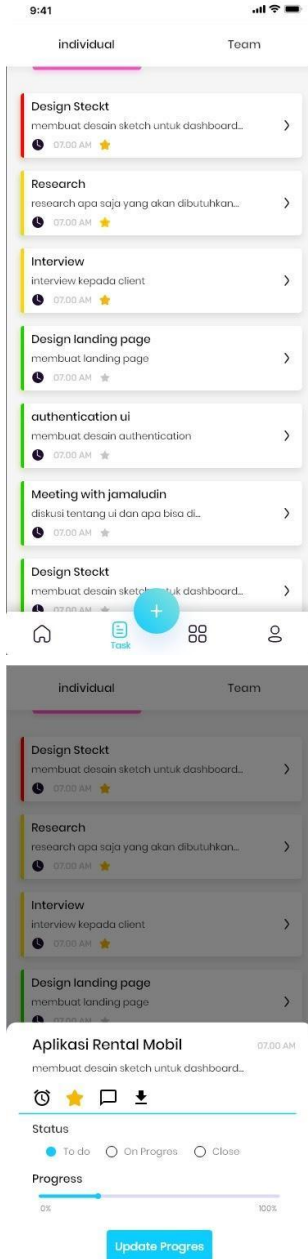
Gambar 5. Tampilan Menu Create Project

Gambar 6. Tampilan Menu Create Task

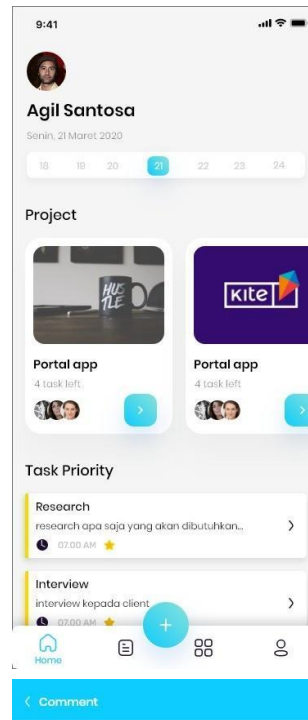
Pada gambar 5 tampilan create project, tata letak form disusun secara berurutan dari yang umum seperti nama project dan deskripsinya hingga kolom lain yang bersifat opsional seperti melampirkan file. Dibuat berturut, supaya pengguna mengisi mulai dari yang terpenting, selain itu juga untuk memudahkan dalam melakukan validasi jika terdapat kolom yang belum terisi.

Pada gambar 6 saat menambahkan task pada sebuah project tidak jauh berbeda dengan tampilan create project. Yang membedakan adalah pada menu ini pengguna (manajer) diminta untuk memilih pengguna lainnya (karyawan) yang akan diberikan tugas. Pemilihan

karyawan dibuat *list* secara horizontal agar tidak memakan banyak ruang, dan dirasa lebih nyaman ketika geser-geser kiri ke kanan. Sambil tetap melihat nama tugas yang diberikan, agar tidak salah memilih orang yang ditugaskan.



Gambar 7. Tampilan Menu Project & Task



Gambar 9. Tampilan Menu Create Project



Gambar 10. Tampilan Menu Discuss Progress

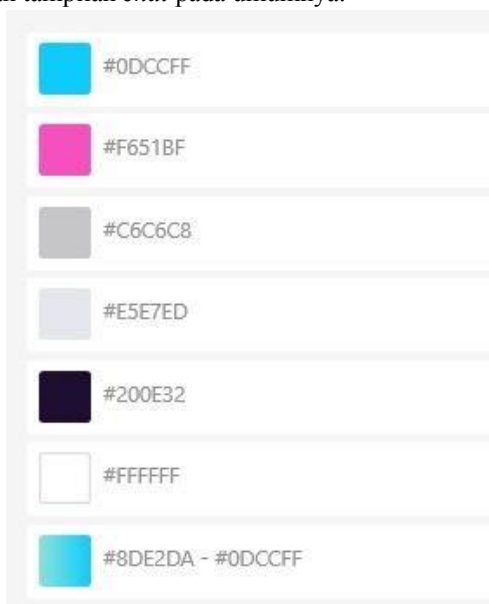
Gambar 8. Tampilan Menu Report Progress

Pada menu untuk menampilkan tugas yang diberikan, dibedakan menjadi 2 antara individu dan *team* dengan dipisahkan oleh menu *tab* seperti gambar 7. Pada daftar tugas yang disajikan juga divariasikan dengan warna yang menggambarkan status dari tugas tersebut. Semisal warna merah yang berarti tugas ditolak oleh manajer, warna kuning tugas sedang on *progress*, warna hijau tugas sudah selesai. Juga terdapat penanda bintang, yang mengartikan tugas tersebut memiliki prioritas untuk dikerjakan.

Untuk melihat detail tugas pada gambar 8, dengan memilih salah satunya dan akan ditampilkan dengan gaya

bottom sheet pop up. Sangat *simple* dan tidak memakan banyak perpindahan antar *layout*, dan lebih *friendly* untuk digunakan. Pada menu ini, digunakan karyawan untuk melakukan pelaporan dari *progress* tugas yang sedang dikerjakan. Dan juga terdapat menu untuk melakukan *action* lainnya, seperti *chat*/diskusi tugas, dan *download* lampiran pengarahan tugas jika ada.

Menu utama pada gambar 9 untuk melihat *project/task* yang sudah berdekatan dengan *deadline* ataupun berdasarkan hari yang dipilih. Pada tampilan ini dibuat untuk mengetahui jumlah tugas yang belum selesai dari sebuah *project*, dan siapa saja yang belum mengerjakannya. Agar memudahkan proses pemantauan tugas oleh manajer. Selanjutnya terdapat menu *chat*/diskusi pada gambar 10, tampilan kurang lebih sama dengan tampilan *chat* pada umumnya.



Gambar 11. Warna dasar Aplikasi To-do List.

Warna dasar seperti ditunjukkan gambar 11 yang digunakan pada aplikasi ini meliputi warna biru muda sebagai warna utama, warna merah muda/pink sebagai warna *accent*, warna abu-abu/silver/putih sebagai perpaduan warna *background*, warna hitam untuk warna *font*, dan terdapat warna *gradient* untuk *background* di beberapa *item view*. Serta ketiga warna untuk menggambarkan status dari sebuah tugas.

D. IMPLEMENTASI SISTEM

Implementasi sistem yang telah dirancang berupa *scaling mockup* menjadi sebuah *layout android* dengan melakukan *coding* pada bahasa *markup XML*. Selain itu juga membuat *application programming interface (API)* untuk menghubungkan *database* pada *server* yang telah dirancang dengan *client* yaitu aplikasi *android*. Bahasa yang digunakan adalah *Native Java* untuk membuat aplikasi *android* beserta beberapa *library* bantuan seperti *Retrofit2* untuk melakukan *request* ke *server*; sedangkan pada *server* menggunakan bahasa PHP dengan *framework Codeigniter*.

IV. PENUTUP

Dalam perancangan aplikasi *To-do List* dengan menggunakan metode pengembangan sistem *Software Development Life Cycle* yang sesuai dengan kebutuhan pengguna untuk menjawab permasalahan yang ada. Disimpulkan sistem sudah layak untuk masuk ke dalam proses pembuatan/*developing* dan juga diuji coba kepada pengguna di PT. Pupuk Kujang Cikampek. Hal ini didukung dengan rancangan sistem yang sudah sangat kompleks, dengan rancangan kebutuhan sistem seperti *Use Case Diagram*, *DFD*, dan *Database Diagram*, beserta dengan rancangan tampilan dari aplikasi yang diinginkan melalui *mockup*.

V. REFERENSI

- [1] Almuttaqi, A. Ibrahim. "Kekacauan respons terhadap Covid-19 di Indonesia." *The Insights* 13 (2020).
- [2] Purwanto, Agus. "Studi eksplorasi dampak Work from Home (WFH) terhadap kinerja guru selama pandemi COVID-19." *EduPsyCouns: Journal of Education, Psychology and Counseling* 2.1 (2020): 92-100.
- [3] Sawitri, Dara. "Penggunaan google meet untuk work from home di era pandemi coronavirus disease 2019 (Covid-19)." *Prioritas: Jurnal Pengabdian Kepada Masyarakat* 2.01 (2020): 13-21.
- [4] Hudani, N. "Proses komunikasi internal pada sekretariat daerah kota bandung dalam menunjang kelancaran kegiatan walikota bandung" (2018).
- [5] Susanto Anna Dara Andriana, Rani. "Perbandingan model waterfall dan prototyping untuk pengembangan sistem informasi." *Majalah Ilmiah UNIKOM* (2016).
- [6] Majid, Nuur Wachid Abdul, et al. "Progress Report of Cyber Society v1. 0 Development as A Learning Media for Indonesian Society to Support EFA." *Int. J. Eng. Pedagog.* 10.4 (2020): 133-145.