**CATATAN HARIAN TIM PROYEK MANPRO**

| Nama | : | Albert P Simanjuntak |
| --- | --- | --- |
| Peran | : | BackEnd Android |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Tanggal** | **Kegiatan** | **Hasil** | **Durasi (jam)** | **Approval oleh SQA** |
| 1. | 10 Februari 2020 | Melakukan brainstorming | Terlaksana | 13.00 – 15.00 |  |
| 2. | 10 Februari 2020 | Ekplorasi penggunaan Firebase dalam proyek | Terlaksana | 08.00 - 10.00 | *√* |
| 3. | 12 Februari 2020 | Latihan dalam penggunaan MVVM dalam proyek | Terlaksana | 17.00 – 19.00 | .. |
| 4. | 14 Februari 2020 | Latihan Repository dan Live Data | Terlaksana | 20.00 – 22.25 | .. |
| 5. | 15 Februari 2020 | Ekplorasi Layout yang akan digunakan | Terlaksana | 08.00-10.00 | .. |
| 6. |  |  |  |  | .. |
| 7. |  |  |  |  | .. |
| 8. |  |  |  |  | .. |

**Total Jam: 10.5 Jam**

**Dibuat oleh Disetujui oleh**

**SQA Proyek Manpro**

**(Albert P Simanjuntak) (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)**

*\*Nama Individu*

Lampiran:

1. Kegiatan 10 Februari 2020

Nama Kegiatan : Melakukan brainstorming

1. Kegiatan 10 Februari 2020

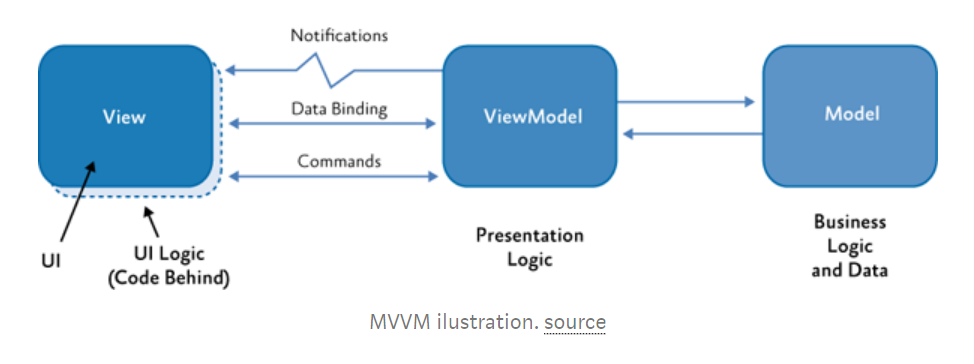
Nama Kegiatan : Ekplorasi penggunaan Firebase dalam proyek

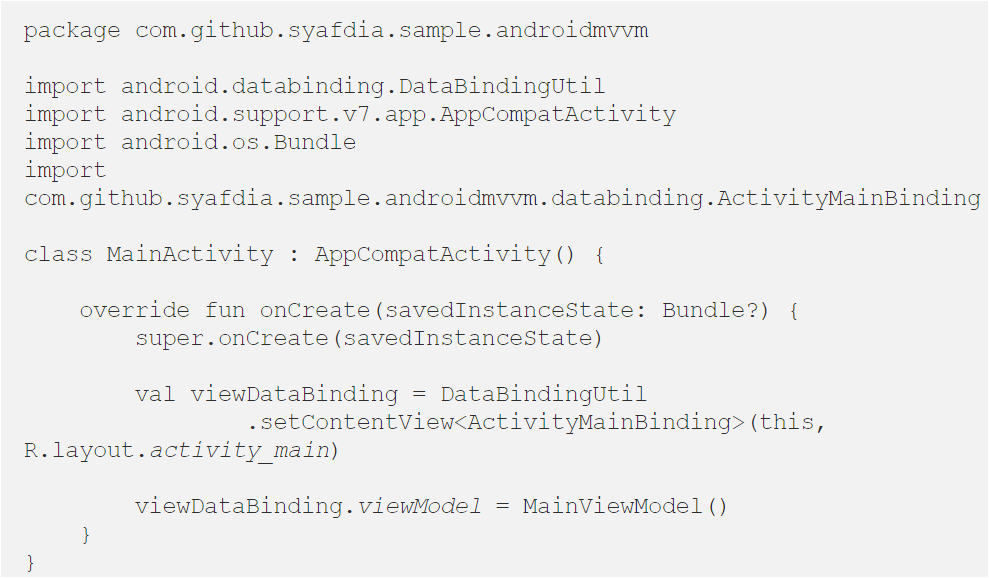


1. Kegiatan 12 Februari 2020

Nama Kegiatan : Latihan dalam penggunaan MVVM dalam proyek

[Model View ViewModel (MVVM)](https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93viewmodel) merupakan salah satu [*architectural pattern*](https://en.wikipedia.org/wiki/Architectural_pattern)*yang dipakai dalam pengembangan perangkat lunak berbasis Graphical User Interface (GUI), arsitektur ini memisahkan business logic*dengan*GUI.*Pada Android SDK sendiri, telah tersedia *library*[Data Binding](https://developer.android.com/topic/libraries/data-binding/index.html) dari API level 7+.





1. Kegiatan 14 Februari 2020

Nama Kegiatan : Latihan Repository dan Live Data

1. **Repository**  
   Repository pattern adalah sebuah *pattern*atau pola pendekatan untuk *mengatur sumber data* sehingga aplikasi dapat berjalan walaupun dalam keadaan online maupun offline. Ketika online, aplikasi akan mengambil data API yang berasal dari *network*, sedangkan ketika offline aplikasi akan menampilkan data dari *local database* yang sudah disimpan sebelumnya. Sehingga aplikasi dapat terus diakses dalam kondisi apa pun, tentunya ini akan meningkatkan kenyamanan pengguna aplikasi Anda.
2. **LiveData**  
   LiveData adalah *observable* *class* *holder* yang akan memberikan *pemberitahuan jika terjadi perubahan data*. Tidak seperti kelas *observable* lainnya, LiveData merupakan*Lifecycle-Aware*. Ini artinya LiveData dapat merespon perubahan yang ada di Lifecycle (Activity, Fragment, atau Service).  LiveData hanya akan memperbarui komponen pengamat atau *observers* ketika *Lifecycle* aplikasi dalam keadaan aktif.
3. Kegiatan 15 Februari 2020

Nama Kegiatan : Ekplorasi Layout yang akan digunakan

