1. An Android "ToDo" application is designed to assist users in managing and organizing their tasks efficiently. The main goal of this application is to provide an efficient and structured way for users to track, add, and complete their daily tasks. Fitur’s in this application are adding new tasks, Marking Completed Tasks, Filtering Completed or Uncompleted Tasks, and Notifications and Reminders. This application have some architectural for example data,notification,setting,ui,utils.
2. Bagian yang saya rasa paling sulit ialah menambahkan AddTaskViewmodel dan menambahkan task ke database dibandingkan yang lain, karena disitu perlu menambahkan fungsi didalam AddTaskViewmodel. Dan pembuatan fitur tambah task di AddTaskActivity. Solusi mengenai hal tersebut dengan cara implementasi fungsi yang telah dibuat di AddTaskViewmodel, lalu membuat tambah task di AddTaskActivity dengan cara a when text error ketika tombol di toast bila data belum di isi, else menambahkan task dari fungsi di AddTaskViewmodel yang mana perlu mengisi title, description, dan due date.
3. The components used in developing a ToDo app involve several key parts of the Android architecture. Here are some components that may be utilized:
   1. Activity: Activity is the primary component representing a screen with a user interface. In the ToDo app, there might be Activities for displaying the task list, adding new tasks, and editing tasks.
   2. Layouts: Layouts are used to organize the user interface layout. XML is used to define the layout structure and is typically found in the res/layout/ directory.
   3. RecyclerView: RecyclerView is used to display dynamic lists of items. In the ToDo app, it can be employed to showcase the list of tasks.
   4. Adapter: The Adapter is used to connect data with the RecyclerView. The Adapter can be found within the RecyclerView implementation.
   5. Room Database: To store and manage task data, either SQLite Database or Room Database may be used. Room is located in the android.arch.persistence.room package.
   6. Intent: Intents are used for transitioning between Activities. In the ToDo app, Intents might be used to open the Add Task Activity or Edit Task Activity.
   7. SharedPreferences: To store application settings or other simple data, SharedPreferences can be utilized. It can be found in the android.content package.
   8. **SharedPreferences:**
      1. The ToDo app will store the user-selected reminder time using SharedPreferences.
      2. After the user selects the reminder time through the app's user interface, save that time using SharedPreferences.
   9. **Repository to Retrieve Data:**
      1. Develop a Repository class to manage access to the application's data.
      2. The Repository will be responsible for storing and retrieving task information, including the reminder time, from a database or another data source.
   10. **WorkManager (WorkRequest, Worker):**
       1. Create a Worker class that will perform the reminder task. Extend the Worker class and implement the **doWork()** method.
       2. Inside the **doWork()** method, use the Repository to retrieve task data that requires a reminder.
       3. Create a WorkRequest object using OneTimeWorkRequest or PeriodicWorkRequest, depending on the application's needs.
       4. Schedule the job using WorkManager by adding the WorkRequest to the queue.
   11. **Notification Manager:**
       1. In the Worker class, after obtaining the task data, use the Notification Manager to display the reminder notification.
       2. Customize the notification with relevant information, such as the title and content of the task that needs to be remembered.
   12. **Pending Intent:**
       1. When the notification is tapped, create a PendingIntent to handle the action to be taken.
       2. The PendingIntent can be used to open an Activity displaying task details or perform specific actions according to your application's needs.
4. Vdfasbvdfab
5. Fadbfdab
6. Dfabvadfb
7. Bdfabfdab
8. Bfdab
9. Adf
10. fddab