

BUKU KERJA PRAKTEK MAHASISWA



WORKSHOP APLIKASI MOBILE FRAMEWORK

SEMESTER 3

Tim Penyusun:

I Gede Wiryawan, S.Kom., M.Kom.

Arvita Agus Kurniasari, S.ST., M.Tr.Kom.

**D4 TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI**

POLITEKNIK NEGERI JEMBER

2022

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET DAN TEKNOLOGI
POLITEKNIK NEGERI JEMBER

LEMBAR PENGESAHAN
WORKSHOP APLIKASI MOBILE FRAMEWORK

Penulis 1

I Gede Wiryawan, S.Kom., M.Kom.
NIP. 19880117 201903 1 008

Penulis 2

Arvita Agus Kurniasari, S.ST., M.Tr.Kom.
NIP. 19930831 202103 2 001

Menyetujui,
Ketua Jurusan

Mengetahui,
Koordinator Program Studi

Hendra Yufit Riskiawan, S.Kom., M.Cs.
NIP. 19830203 200604 1 003

Trismayanti Dwi P., S.Kom., M.Cs.
NIP. 19900227 201803 2 001

KATA PENGANTAR

Puji Syukur kehadirat Tuhan Yang Maha Esa karena atas limpahan rahmat-Nya sehingga kami dapat menyelesaikan Buku Kerja Praktim Mahasiswa (BKPM) Workshop Aplikasi Mobile. Materi Workshop terdiri dari pengenalan github, pengenalan android studio, layout, activity, fragment, intent, manajemen file, shared preferences, GPS dan akses sensor, JSON, pengenalan retrofit, volley, get, **post** dan project final. BKPM ini disusun berdasarkan metode *Student Center Learning* yaitu menempatkan mahasiswa sebagai pusat kegiatan belajar. BKPM ini terdiri dari Pokok Bahasan, Acara Praktikum, Tempat, Alokasi Waktu, Capaian Pembelajaran Mata Kuliah (CPMK), Indikator, Dasar Teori, Alat dan Bahan, Prosedur Kerja, Hasil dan Pembahasan, Kesimpulan, Rubrik Penilaian.

Kami menyadari masih banyak kekurangan dalam penyusunan BKPM ini. Oleh karena itu, kami sangat mengharapkan kritik dan saran demi perbaikan dan kesempurnaan modul ini. Kami mengucapkan terima kasih kepada berbagai pihak yang telah membantu proses penyelesaian BKPM ini. Semoga BKPM ini dapat bermanfaat bagi kita semua.

Jember, 1 September 2021

Tim Penulis

DAFTAR ISI

| | |
|--|-----|
| LEMBAR PENGESAHAN | ii |
| KATA PENGANTAR | iii |
| DAFTAR ISI..... | iv |
| ACARA 1 : INTRODUCTION DAN WORKFLOW PADA GIT, FLUTTER & DART | 1 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 1 |
| b. Indikator..... | 1 |
| c. Dasar Teori..... | 1 |
| d. Alat dan Bahan..... | 2 |
| f. Hasil dan Pembahasan | 2 |
| g. Kesimpulan | 2 |
| h. Rubrik Penilaian..... | 2 |
| ACARA 2 : PENGENALAN DAN WORKFLOW PADA GIT, FLUTTER & DART | 3 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 3 |
| b. Indikator..... | 3 |
| c. Dasar Teori..... | 3 |
| d. Alat dan Bahan..... | 3 |
| e. Prosedur Kerja | 3 |
| f. Hasil dan Pembahasan | 13 |
| g. Kesimpulan | 14 |
| h. Rubrik Penilaian..... | 14 |
| ACARA 3 : PENGENALAN DAN INSTALASI FLUTTER, FLUTTER & DART | 15 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 15 |
| b. Indikator..... | 15 |
| c. Dasar Teori..... | 15 |
| d. Alat dan Bahan..... | 15 |
| e. Prosedur Kerja | 15 |
| f. Hasil dan Pembahasan | 42 |
| g. Kesimpulan | 42 |
| h. Rubrik Penilaian..... | 42 |
| ACARA 4 : PENGENALAN DAN WORKFLOW PADA GIT, FLUTTER & DART | 43 |

| | | |
|----|--|-----------|
| a. | Capaian Pembelajaran Mata Kuliah (CPMK)..... | 43 |
| b. | Indikator..... | 43 |
| c. | Dasar Teori..... | 43 |
| d. | Alat dan Bahan..... | 43 |
| e. | Prosedur Kerja | 43 |
| f. | Hasil dan Pembahasan | 49 |
| g. | Kesimpulan | 49 |
| h. | Rubrik Penilaian..... | 49 |
| | ACARA 5 : PEMROGRAMAN DART I..... | 50 |
| a. | Capaian Pembelajaran Mata Kuliah (CPMK)..... | 50 |
| b. | Indikator..... | 50 |
| c. | Dasar Teori..... | 50 |
| d. | Alat dan Bahan..... | 54 |
| e. | Prosedur Kerja | 54 |
| f. | Hasil dan Pembahasan | 55 |
| g. | Kesimpulan | 55 |
| h. | Rubrik Penilaian..... | 56 |
| | ACARA 6 : PEMROGRAMAN DART I..... | 57 |
| a. | Capaian Pembelajaran Mata Kuliah (CPMK)..... | 57 |
| b. | Indikator..... | 57 |
| c. | Dasar Teori..... | 57 |
| d. | Alat dan Bahan..... | 60 |
| e. | Prosedur Kerja | 60 |
| f. | Hasil dan Pembahasan | 60 |
| g. | Kesimpulan | 60 |
| h. | Rubrik Penilaian..... | 60 |
| | ACARA 7 : PEMROGRAMAN DART I..... | 61 |
| a. | Capaian Pembelajaran Mata Kuliah (CPMK)..... | 61 |
| b. | Indikator..... | 61 |
| c. | Dasar Teori..... | 61 |
| d. | Alat dan Bahan..... | 63 |
| e. | Prosedur Kerja | 63 |

| | | |
|----|--|-----------|
| f. | Hasil dan Pembahasan | 64 |
| g. | Kesimpulan | 64 |
| h. | Rubrik Penilaian..... | 64 |
| | ACARA 8 : PEMROGRAMAN DART I..... | 65 |
| a. | Capaian Pembelajaran Mata Kuliah (CPMK)..... | 65 |
| b. | Indikator..... | 65 |
| c. | Dasar Teori..... | 65 |
| d. | Alat dan Bahan..... | 65 |
| e. | Prosedur Kerja | 65 |
| f. | Hasil dan Pembahasan | 69 |
| g. | Kesimpulan | 70 |
| h. | Rubrik Penilaian..... | 70 |
| | ACARA 9 : PEMROGRAMAN DART II | 71 |
| a. | Capaian Pembelajaran Mata Kuliah (CPMK)..... | 71 |
| b. | Indikator..... | 71 |
| c. | Dasar Teori..... | 71 |
| d. | Alat dan Bahan..... | 71 |
| e. | Prosedur Kerja | 72 |
| f. | Hasil dan Pembahasan | 73 |
| g. | Kesimpulan | 73 |
| h. | Rubrik Penilaian..... | 73 |
| | ACARA 10 : PEMROGRAMAN DART II | 74 |
| a. | Capaian Pembelajaran Mata Kuliah (CPMK)..... | 74 |
| b. | Indikator..... | 74 |
| c. | Dasar Teori..... | 74 |
| d. | Alat dan Bahan..... | 76 |
| e. | Prosedur Kerja | 76 |
| f. | Hasil dan Pembahasan | 79 |
| g. | Kesimpulan | 79 |
| h. | Rubrik Penilaian..... | 79 |
| | ACARA 11 : PEMROGRAMAN DART II | 80 |
| a. | Capaian Pembelajaran Mata Kuliah (CPMK)..... | 80 |

| | | |
|----|---|-----------|
| b. | Indikator | 80 |
| c. | Dasar Teori..... | 80 |
| d. | Alat dan Bahan..... | 81 |
| e. | Prosedur Kerja | 81 |
| f. | Hasil dan Pembahasan | 82 |
| g. | Kesimpulan | 82 |
| h. | Rubrik Penilaian..... | 82 |
| | ACARA 12 : PEMROGRAMAN DART II | 83 |
| a. | Capaian Pembelajaran Mata Kuliah (CPMK)..... | 83 |
| c. | Dasar Teori..... | 83 |
| d. | Alat dan Bahan..... | 87 |
| e. | Prosedur Kerja | 87 |
| f. | Hasil dan Pembahasan | 88 |
| g. | Kesimpulan | 88 |
| h. | Rubrik Penilaian..... | 88 |
| | ACARA 13 : MOCK UP & WIDGET CORE COMPONENT | 89 |
| a. | Capaian Pembelajaran Mata Kuliah (CPMK)..... | 89 |
| b. | Indikator | 89 |
| c. | Dasar Teori..... | 89 |
| d. | Alat dan Bahan..... | 90 |
| e. | Prosedur Kerja | 90 |
| f. | Hasil dan Pembahasan | 94 |
| g. | Kesimpulan | 94 |
| h. | Rubrik Penilaian..... | 94 |
| | ACARA 14 : MOCK UP & WIDGET CORE COMPONENT | 95 |
| a. | Capaian Pembelajaran Mata Kuliah (CPMK)..... | 95 |
| b. | Indikator | 95 |
| c. | Dasar Teori..... | 95 |
| d. | Alat dan Bahan..... | 96 |
| e. | Prosedur Kerja | 96 |
| f. | Hasil dan Pembahasan | 96 |
| g. | Kesimpulan | 97 |

| | |
|--|------------|
| h. Rubrik Penilaian..... | 97 |
| ACARA 15 : MOCK UP & WIDGET CORE COMPONENT | 98 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 98 |
| b. Indikator..... | 98 |
| c. Dasar Teori..... | 98 |
| d. Alat dan Bahan..... | 104 |
| e. Prosedur Kerja | 104 |
| f. Hasil dan Pembahasan | 104 |
| g. Kesimpulan | 104 |
| h. Rubrik Penilaian..... | 105 |
| ACARA 16 : MOCK UP & WIDGET CORE COMPONENT | 106 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 106 |
| b. Indikator..... | 106 |
| c. Dasar Teori..... | 106 |
| d. Alat dan Bahan..... | 108 |
| e. Prosedur Kerja | 109 |
| f. Hasil dan Pembahasan | 115 |
| g. Kesimpulan | 115 |
| h. Rubrik Penilaian..... | 115 |
| ACARA 17 : FLUTTER STYLING, NAVIGATION, & BUTTON TAB NAVIGATION | 116 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 116 |
| b. Indikator..... | 116 |
| c. Dasar Teori..... | 116 |
| d. Alat dan Bahan..... | 116 |
| e. Prosedur Kerja | 116 |
| f. Hasil dan Pembahasan | 128 |
| g. Kesimpulan | 128 |
| h. Rubrik Penilaian..... | 128 |
| ACARA 18 : FLUTTER STYLING, NAVIGATION, & BUTTON TAB NAVIGATION | 129 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 129 |
| b. Indikator..... | 129 |
| c. Dasar Teori..... | 129 |

| | | |
|----|---|------------|
| d. | Alat dan Bahan..... | 130 |
| e. | Prosedur Kerja | 130 |
| f. | Hasil dan Pembahasan | 132 |
| g. | Kesimpulan | 132 |
| h. | Rubrik Penilaian..... | 132 |
| | ACARA 19 : FLUTTER STYLING, NAVIGATION, & BUTTON TAB NAVIGATION..... | 133 |
| a. | Capaian Pembelajaran Mata Kuliah (CPMK)..... | 133 |
| b. | Indikator..... | 133 |
| c. | Dasar Teori..... | 133 |
| d. | Alat dan Bahan..... | 133 |
| e. | Prosedur Kerja | 133 |
| f. | Hasil dan Pembahasan | 137 |
| g. | Kesimpulan | 137 |
| a. | Rubrik Penilaian..... | 137 |
| | ACARA 20 : FLUTTER STYLING, NAVIGATION, & BUTTON TAB NAVIGATION..... | 139 |
| a. | Capaian Pembelajaran Mata Kuliah (CPMK)..... | 139 |
| b. | Indikator..... | 139 |
| c. | Dasar Teori..... | 139 |
| d. | Alat dan Bahan..... | 141 |
| e. | Prosedur Kerja | 141 |
| f. | Hasil dan Pembahasan | 141 |
| g. | Kesimpulan | 142 |
| h. | Rubrik Penilaian..... | 142 |
| | ACARA 21 : GRADIENT, PAGEVIEW, DROPODOWN | 143 |
| a. | Capaian Pembelajaran Mata Kuliah (CPMK)..... | 143 |
| b. | Indikator..... | 143 |
| c. | Dasar Teori..... | 143 |
| d. | Alat dan Bahan..... | 144 |
| e. | Prosedur Kerja | 144 |
| f. | Hasil dan Pembahasan | 145 |
| g. | Kesimpulan | 145 |
| h. | Rubrik Penilaian..... | 145 |

| | |
|---|-----|
| ACARA 22 : GRADIENT, PAGEVIEW, DROPDOWN | 146 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 146 |
| b. Indikator..... | 146 |
| c. Dasar Teori..... | 146 |
| d. Alat dan Bahan..... | 146 |
| e. Prosedur Kerja | 146 |
| f. Hasil dan Pembahasan | 150 |
| g. Kesimpulan | 150 |
| h. Rubrik Penilaian..... | 150 |
| ACARA 23 : GRADIENT, PAGEVIEW, DROPDOWN | 152 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 152 |
| b. Indikator..... | 152 |
| c. Dasar Teori..... | 152 |
| d. Alat dan Bahan..... | 152 |
| e. Prosedur Kerja | 152 |
| f. Hasil dan Pembahasan | 155 |
| g. Kesimpulan | 155 |
| h. Rubrik Penilaian..... | 155 |
| ACARA 24 : GRADIENT, PAGEVIEW, DROPDOWN | 156 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 156 |
| b. Indikator..... | 156 |
| c. Dasar Teori..... | 156 |
| d. Alat dan Bahan..... | 156 |
| e. Prosedur Kerja | 156 |
| f. Hasil dan Pembahasan | 156 |
| g. Kesimpulan | 157 |
| h. Rubrik Penilaian..... | 157 |
| ACARA 25 : FORM | 158 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 158 |
| b. Indikator..... | 158 |
| c. Dasar Teori..... | 158 |
| d. Alat dan Bahan..... | 163 |

| | |
|---|------------|
| e. Prosedur Kerja | 163 |
| f. Hasil dan Pembahasan | 165 |
| g. Kesimpulan | 165 |
| h. Rubrik Penilaian..... | 165 |
| ACARA 26 : FORM | 167 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 167 |
| b. Indikator..... | 167 |
| c. Dasar Teori..... | 167 |
| d. Alat dan Bahan..... | 170 |
| e. Prosedur Kerja | 170 |
| f. Hasil dan Pembahasan | 173 |
| g. Kesimpulan | 173 |
| h. Rubrik Penilaian..... | 173 |
| ACARA 27 : FORM | 174 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 174 |
| b. Indikator..... | 174 |
| c. Dasar Teori..... | 174 |
| d. Alat dan Bahan..... | 174 |
| e. Prosedur Kerja | 174 |
| f. Hasil dan Pembahasan | 179 |
| g. Kesimpulan | 179 |
| h. Rubrik Penilaian..... | 179 |
| ACARA 28 : FORM | 181 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 181 |
| b. Indikator..... | 181 |
| c. Dasar Teori..... | 181 |
| d. Alat dan Bahan..... | 181 |
| e. Prosedur Kerja | 181 |
| f. Hasil dan Pembahasan | 181 |
| g. Kesimpulan | 181 |
| h. Rubrik Penilaian..... | 182 |
| ACARA 29 : STATE, ROUTE, & DEPENDENCY MANAGEMENT | 183 |

| | | |
|----|---|------------|
| a. | Capaian Pembelajaran Mata Kuliah (CPMK)..... | 183 |
| b. | Indikator..... | 183 |
| c. | Dasar Teori..... | 183 |
| d. | Alat dan Bahan..... | 185 |
| e. | Prosedur Kerja | 185 |
| f. | Hasil dan Pembahasan | 186 |
| g. | Kesimpulan | 186 |
| h. | Rubrik Penilaian..... | 186 |
| | ACARA 30 : STATE, ROUTE, & DEPENDENCY MANAGEMENT | 187 |
| a. | Capaian Pembelajaran Mata Kuliah (CPMK)..... | 187 |
| b. | Indikator..... | 187 |
| c. | Dasar Teori..... | 187 |
| d. | Alat dan Bahan..... | 188 |
| e. | Prosedur Kerja | 188 |
| f. | Hasil dan Pembahasan | 189 |
| g. | Kesimpulan | 190 |
| h. | Rubrik Penilaian..... | 190 |
| | ACARA 31 : STATE, ROUTE, & DEPENDENCY MANAGEMENT | 191 |
| a. | Capaian Pembelajaran Mata Kuliah (CPMK)..... | 191 |
| b. | Indikator..... | 191 |
| c. | Dasar Teori..... | 191 |
| d. | Alat dan Bahan..... | 193 |
| e. | Prosedur Kerja | 193 |
| f. | Hasil dan Pembahasan | 198 |
| g. | Kesimpulan | 199 |
| h. | Rubrik Penilaian..... | 199 |
| | ACARA 32 : STATE, ROUTE, & DEPENDENCY MANAGEMENT | 200 |
| a. | Capaian Pembelajaran Mata Kuliah (CPMK)..... | 200 |
| b. | Indikator..... | 200 |
| c. | Dasar Teori..... | 200 |
| d. | Alat dan Bahan..... | 201 |
| e. | Prosedur Kerja | 201 |

| | |
|---|------------|
| f. Hasil dan Pembahasan | 202 |
| g. Kesimpulan | 202 |
| h. Rubrik Penilaian..... | 202 |
| ACARA 33 : REST API | 203 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 203 |
| b. Indikator..... | 203 |
| c. Dasar Teori..... | 203 |
| d. Alat dan Bahan..... | 217 |
| e. Prosedur Kerja | 217 |
| f. Hasil dan Pembahasan | 220 |
| g. Kesimpulan | 220 |
| h. Rubrik Penilaian..... | 220 |
| ACARA 34 : REST API | 221 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 221 |
| b. Indikator..... | 221 |
| c. Dasar Teori..... | 221 |
| d. Alat dan Bahan..... | 221 |
| e. Prosedur Kerja | 221 |
| f. Hasil dan Pembahasan | 226 |
| g. Kesimpulan | 226 |
| h. Rubrik Penilaian..... | 226 |
| ACARA 35 : REST API | 228 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 228 |
| b. Indikator..... | 228 |
| c. Dasar Teori..... | 228 |
| d. Alat dan Bahan..... | 232 |
| e. Prosedur Kerja | 232 |
| f. Hasil dan Pembahasan | 240 |
| g. Kesimpulan | 240 |
| h. Rubrik Penilaian..... | 240 |
| ACARA 36 : REST API | 241 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 241 |

| | | |
|----|--|------------|
| b. | Indikator | 241 |
| c. | Dasar Teori..... | 241 |
| d. | Alat dan Bahan..... | 241 |
| e. | Prosedur Kerja | 241 |
| f. | Hasil dan Pembahasan | 241 |
| g. | Kesimpulan | 241 |
| h. | Rubrik Penilaian..... | 242 |
| | ACARA 37 : AUTHENTICATION | 243 |
| a. | Capaian Pembelajaran Mata Kuliah (CPMK)..... | 243 |
| b. | Indikator..... | 243 |
| c. | Dasar Teori..... | 243 |
| d. | Alat dan Bahan..... | 246 |
| e. | Prosedur Kerja | 246 |
| f. | Hasil dan Pembahasan | 253 |
| g. | Kesimpulan | 253 |
| h. | Rubrik Penilaian..... | 253 |
| | ACARA 38 : AUTHENTICATION | 254 |
| a. | Capaian Pembelajaran Mata Kuliah (CPMK)..... | 254 |
| b. | Indikator..... | 254 |
| c. | Dasar Teori..... | 254 |
| d. | Alat dan Bahan..... | 254 |
| e. | Prosedur Kerja | 254 |
| f. | Hasil dan Pembahasan | 256 |
| g. | Kesimpulan | 256 |
| h. | Rubrik Penilaian..... | 256 |
| | ACARA 39 : AUTHENTICATION | 257 |
| a. | Capaian Pembelajaran Mata Kuliah (CPMK)..... | 257 |
| b. | Indikator..... | 257 |
| c. | Dasar Teori..... | 257 |
| d. | Alat dan Bahan..... | 257 |
| e. | Prosedur Kerja | 257 |
| f. | Hasil dan Pembahasan | 260 |

| | |
|---|------------|
| g. Kesimpulan | 260 |
| h. Rubrik Penilaian..... | 260 |
| ACARA 40 : AUTHENTICATION | 261 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 261 |
| b. Indikator | 261 |
| c. Dasar Teori..... | 261 |
| d. Alat dan Bahan..... | 261 |
| e. Prosedur Kerja | 261 |
| f. Hasil dan Pembahasan | 263 |
| g. Kesimpulan | 263 |
| h. Rubrik Penilaian..... | 263 |
| ACARA 41 : CRUD | 264 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 264 |
| b. Indikator..... | 264 |
| c. Dasar Teori..... | 264 |
| d. Alat dan Bahan..... | 265 |
| e. Prosedur Kerja | 265 |
| f. Hasil dan Pembahasan | 266 |
| g. Kesimpulan | 266 |
| h. Rubrik Penilaian..... | 266 |
| ACARA 42 : CRUD | 268 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 268 |
| b. Indikator..... | 268 |
| c. Dasar Teori..... | 268 |
| d. Alat dan Bahan..... | 268 |
| e. Prosedur Kerja | 268 |
| f. Hasil dan Pembahasan | 270 |
| g. Kesimpulan | 270 |
| h. Rubrik Penilaian..... | 270 |
| ACARA 43 : CRUD | 272 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 272 |
| b. Indikator..... | 272 |

| | | |
|----|--|------------|
| c. | Dasar Teori..... | 272 |
| d. | Alat dan Bahan..... | 272 |
| e. | Prosedur Kerja | 272 |
| f. | Hasil dan Pembahasan | 277 |
| g. | Kesimpulan | 277 |
| h. | Rubrik Penilaian..... | 277 |
| | ACARA 44 : CRUD | 279 |
| a. | Capaian Pembelajaran Mata Kuliah (CPMK)..... | 279 |
| b. | Indikator | 279 |
| c. | Dasar Teori..... | 279 |
| d. | Alat dan Bahan..... | 279 |
| e. | Prosedur Kerja | 279 |
| f. | Hasil dan Pembahasan | 285 |
| g. | Kesimpulan | 285 |
| h. | Rubrik Penilaian..... | 285 |
| | ACARA 45 : PROJECT FINAL | 286 |
| a. | Capaian Pembelajaran Mata Kuliah (CPMK)..... | 290 |
| b. | Indikator | 290 |
| c. | Dasar Teori..... | 290 |
| d. | Alat dan Bahan..... | 293 |
| e. | Prosedur Kerja | 293 |
| f. | Hasil dan Pembahasan | 293 |
| g. | Kesimpulan | 294 |
| h. | Rubrik Penilaian..... | 294 |
| | ACARA 46 : PROJECT FINAL | 295 |
| a. | Capaian Pembelajaran Mata Kuliah (CPMK)..... | 295 |
| b. | Indikator | 295 |
| c. | Dasar Teori..... | 295 |
| d. | Alat dan Bahan..... | 296 |
| e. | Prosedur Kerja | 296 |
| f. | Hasil dan Pembahasan | 297 |
| g. | Kesimpulan | 297 |

| | |
|---|------------|
| h. Rubrik Penilaian..... | 297 |
| ACARA 47 : PROJECT FINAL | 298 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 298 |
| b. Indikator..... | 298 |
| c. Dasar Teori..... | 298 |
| d. Alat dan Bahan..... | 299 |
| e. Prosedur Kerja | 299 |
| f. Hasil dan Pembahasan | 300 |
| g. Kesimpulan | 300 |
| h. Rubrik Penilaian..... | 300 |
| ACARA 48 : PROJECT FINAL | 301 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 301 |
| b. Indikator..... | 301 |
| c. Dasar Teori..... | 301 |
| d. Alat dan Bahan..... | 301 |
| e. Prosedur Kerja | 301 |
| f. Hasil dan Pembahasan | 301 |
| g. Kesimpulan | 301 |
| h. Rubrik Penilaian..... | 302 |
| ACARA 49 : PROJECT FINAL | 303 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 303 |
| b. Indikator..... | 303 |
| c. Dasar Teori..... | 303 |
| d. Alat dan Bahan..... | 305 |
| e. Prosedur Kerja | 305 |
| f. Hasil dan Pembahasan | 305 |
| g. Kesimpulan | 305 |
| h. Rubrik Penilaian..... | 305 |
| ACARA 50 : PROJECT FINAL | 306 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 306 |
| b. Indikator..... | 306 |
| c. Dasar Teori..... | 306 |

| | |
|---|------------|
| d. Alat dan Bahan..... | 307 |
| e. Prosedur Kerja | 307 |
| f. Hasil dan Pembahasan | 307 |
| g. Kesimpulan | 307 |
| h. Rubrik Penilaian..... | 307 |
| ACARA 51 : PROJECT FINAL..... | 308 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 308 |
| b. Indikator..... | 308 |
| c. Dasar Teori..... | 308 |
| d. Alat dan Bahan..... | 308 |
| e. Prosedur Kerja | 308 |
| f. Hasil dan Pembahasan | 308 |
| g. Kesimpulan | 309 |
| h. Rubrik Penilaian..... | 309 |
| ACARA 52 : PROJECT FINAL..... | 310 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 310 |
| b. Indikator..... | 310 |
| c. Dasar Teori..... | 310 |
| d. Alat dan Bahan..... | 310 |
| e. Prosedur Kerja | 310 |
| f. Hasil dan Pembahasan | 310 |
| g. Kesimpulan | 310 |
| h. Rubrik Penilaian..... | 310 |
| ACARA 53 : PROJECT FINAL..... | 286 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 286 |
| b. Indikator..... | 286 |
| c. Dasar Teori..... | 286 |
| d. Alat dan Bahan..... | 288 |
| e. Prosedur Kerja | 288 |
| f. Hasil dan Pembahasan | 289 |
| g. Kesimpulan | 289 |
| h. Rubrik Penilaian..... | 289 |

| | |
|---|-----|
| ACARA 54 : PROJECT FINAL | 312 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 312 |
| b. Indikator..... | 312 |
| c. Dasar Teori..... | 312 |
| d. Alat dan Bahan..... | 312 |
| e. Prosedur Kerja | 312 |
| f. Hasil dan Pembahasan | 312 |
| g. Kesimpulan | 312 |
| h. Rubrik Penilaian..... | 312 |
| ACARA 55 : PROJECT FINAL | 313 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 313 |
| b. Indikator..... | 313 |
| c. Dasar Teori..... | 313 |
| d. Alat dan Bahan..... | 313 |
| e. Prosedur Kerja | 313 |
| f. Hasil dan Pembahasan | 313 |
| g. Kesimpulan | 313 |
| h. Rubrik Penilaian..... | 313 |
| ACARA 54 : PROJECT FINAL | 314 |
| a. Capaian Pembelajaran Mata Kuliah (CPMK)..... | 314 |
| b. Indikator..... | 314 |
| c. Dasar Teori..... | 314 |
| d. Alat dan Bahan..... | 314 |
| e. Prosedur Kerja | 314 |
| f. Hasil dan Pembahasan | 314 |
| g. Kesimpulan | 314 |
| h. Rubrik Penilaian..... | 314 |

ACARA 1 : INTRODUCTION DAN WORKFLOW PADA GIT, FLUTTER & DART

| | |
|----------------------------------|---|
| Pokok Bahasan | : Intro dan Pengenalan Git |
| Acara Praktikum/Pertemuan | : Minggu 1 / 1 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami konsep git
2. Mahasiswa mampu menerapkan git

b. Indikator

Kemampuan mahasiswa dalam memahami konsep dasar git

c. Dasar Teori

Pengenalan Git

Instalasi Git di Windows memang tidak seperti di Linux yang ketik perintah langsung terinstal. Kita harus men-download dulu, kemudian melakukan ritual *next → next → finish*. Tapi dalam ritual tersebut, ada pilihan yang harus diperhatikan agar perintah git dapat dikenali di CMD.

Terdapat tiga kondisi file dalam Git.

1) Modified

Modified adalah kondisi dimana revisi atau perubahan sudah dilakukan, tetapi belum ditandai dan belum disimpan di *version control*. Contohnya pada gambar di atas, ada tiga file HTML yang dalam kondisi *modified*.

2) Staged

Staged adalah kondisi dimana revisi sudah ditandai, tetapi belum disimpan di *version control*. Untuk mengubah kondisi file dari *modified* ke *staged* gunakan perintah git add *nama_file*. Contoh:

```
git add index.html
```

3) Committed

Committed adalah kondisi dimana revisi sudah disimpan di version control. perintah untuk mengubah kondisi file dari *staged* ke *committed* adalah git commit.

d. Alat dan Bahan

1. Kertas HVS
 2. Spidol Besar
 3. Bolpoin
- e. Prosedur Kerja

Pembagian Kelompok Tugas Project

Mahasiswa memperhatikan penjelasan dosen pengampu mata kuliah Workshop Aplikasi Mobile berbasis Android.

1. Mahasiswa menentukan kelompok dengan 4-5 anggota.
2. Membuat resume secara individu tentang konsep dasar dari pengenalan dan workflow git pada kertas folio.
3. Diskusikan dengan kelompok hasil resume yang telah dikerjakan secara individu, dan salin kedalam PPT.
4. Presentasikan hasil membuat resume di depan kelas setiap kelompok secara bergantian.

f. Hasil dan Pembahasan

1. Dokumentasi tugas berupa laporan (definisi, konsep dasar dan pengenalan git).
2. Dokumentasi tugas berupa file ppt.

g. Kesimpulan

Mahasiswa memahami mengenai definisi git, konsep dasar pengenalan dan workflow git.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|-------------|---|-------|-----|-----|-----|
| 1 | Ketepatan waktu dan ketepatan dalam menjelaskan dari tugas ditunjang dengan bukti referensi | 1 | 2 | 3 | (4) |
| 2 | Ketepatan waktu dan ketepatan dalam menjelaskan dari tugas | 1 | 2 | (3) | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat dalam menjelaskan tugas | 1 | (2) | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan dalam menjelaskan tugas | (1) | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 2 : PENGENALAN DAN WORKFLOW PADA GIT, FLUTTER & DART

| | |
|----------------------------------|---|
| Pokok Bahasan | : Instalasi dan Workflow pada Git |
| Acara Praktikum/Pertemuan | : Minggu 1 / 2 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu instalasi Git pada masing-masing PC
2. Mahasiswa mampu melakukan konfigurasi

b. Indikator

Keberhasilan mahasiswa dalam melakukan tahap-tahap instalasi git, sampai dengan mampu melakukan konfigurasi awal untuk akun git setiap individu.

c. Dasar Teori

-

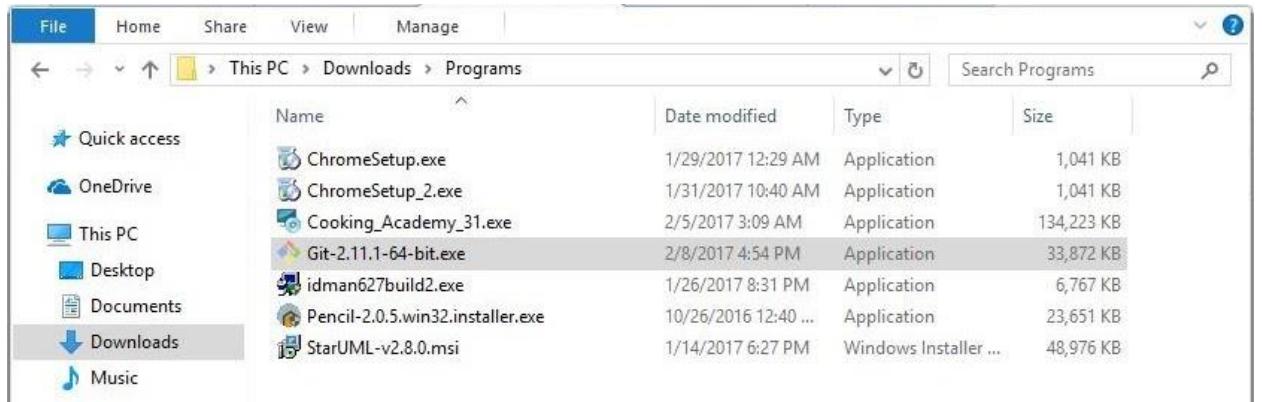
d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

Instalasi Git

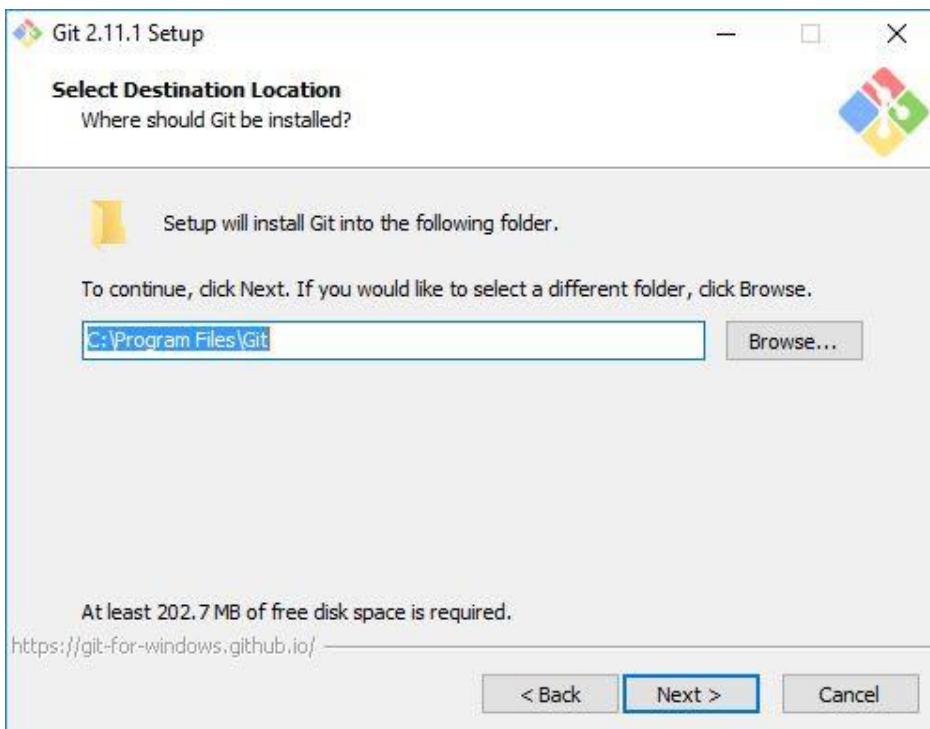
1. Download Git
Silahkan buka website resminya Git (git-scm.com).
2. Kemudian unduh Git sesuai dengan arsitektur komputer yang digunakan. Kalau menggunakan 64bit, unduh yang 64bit. Begitu juga kalau menggunakan 32bit.
3. Setelah file ter-unduh, silahkan klik 2x file instaler Git yang sudah diunduh.



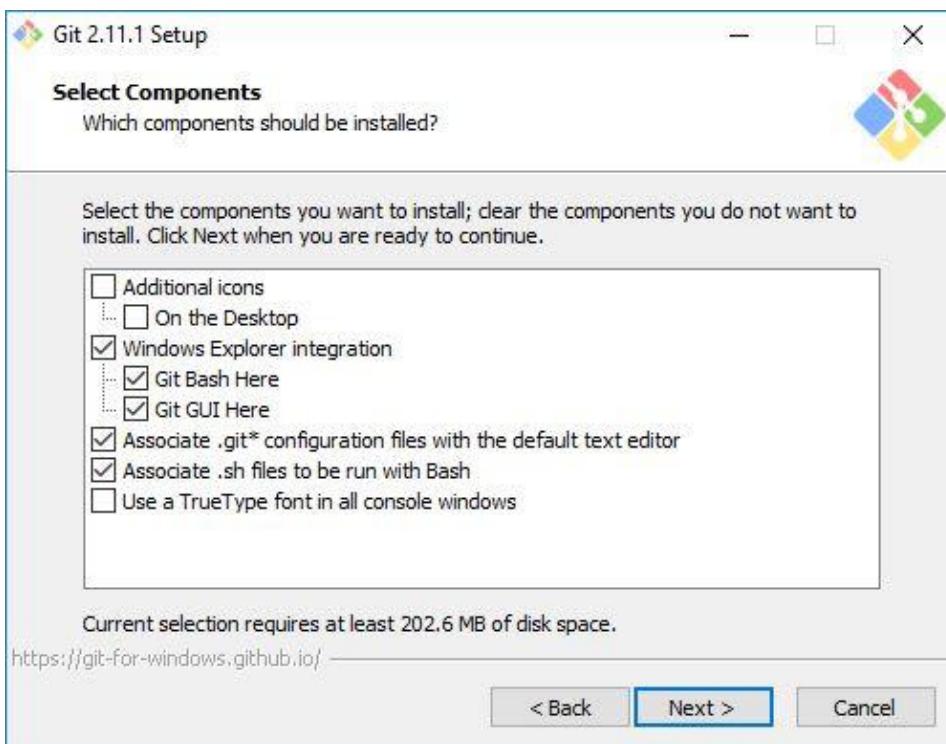
Maka akan muncul infomasi lisensi Git, klik *Next >* untuk melanjutkan.



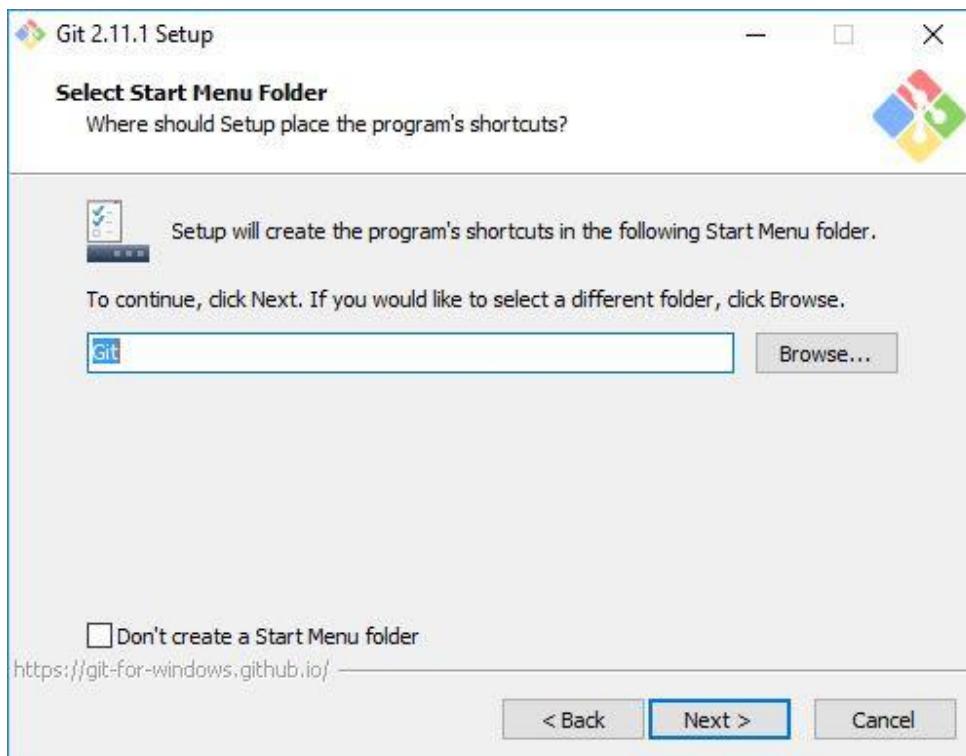
4. Selanjutnya menentukan lokasi instalasi. Biarkan saja apa adanya, kemudian klik *Next >*.



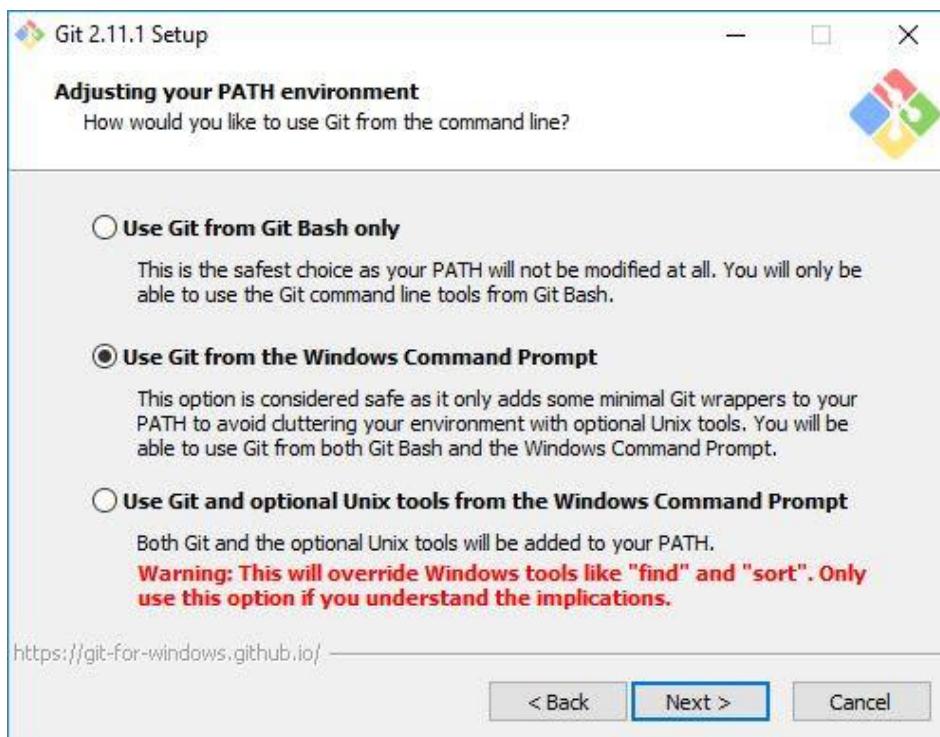
5. Selanjutnya pemilihan komponen, biarkan saja seperti ini kemudian klik *Next >*.



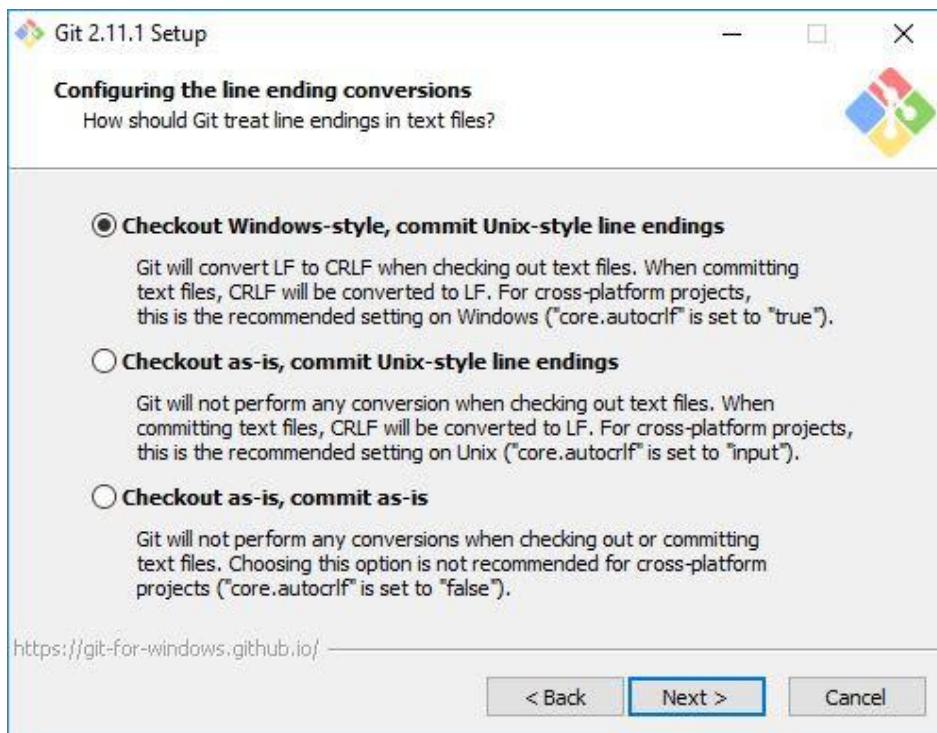
6. Selanjutnya pemilihan direktori start menu, klik *Next >*.



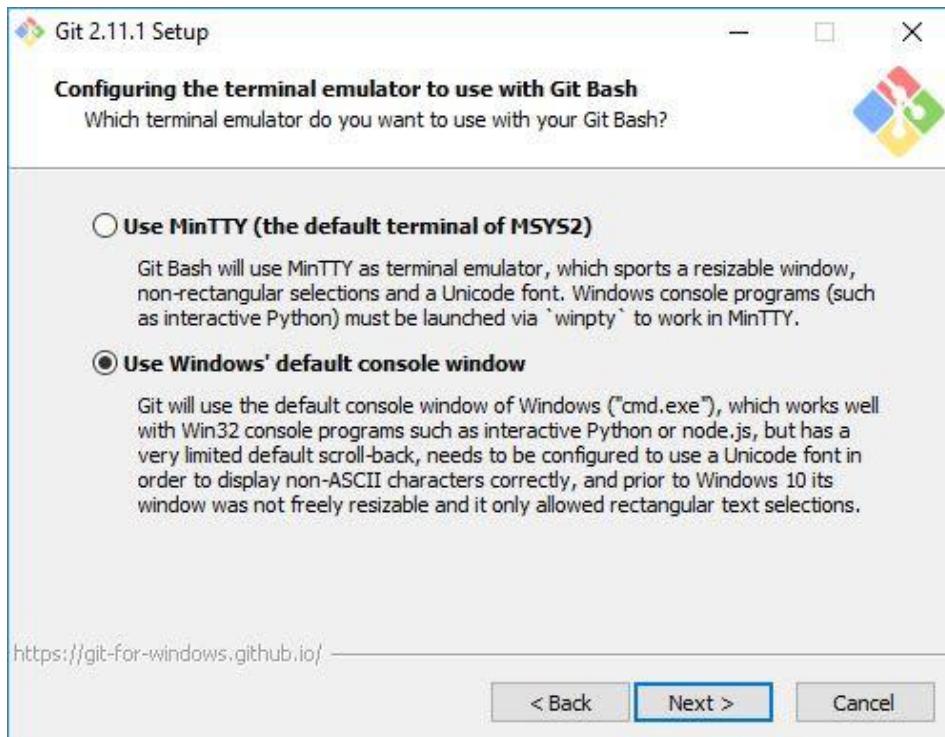
7. Selanjutnya pengaturan *PATH Environment*. Pilih yang tengah agar perintah git dapat di kenali di *Command Prompt* (CMD). Setelah itu klik *Next >*.



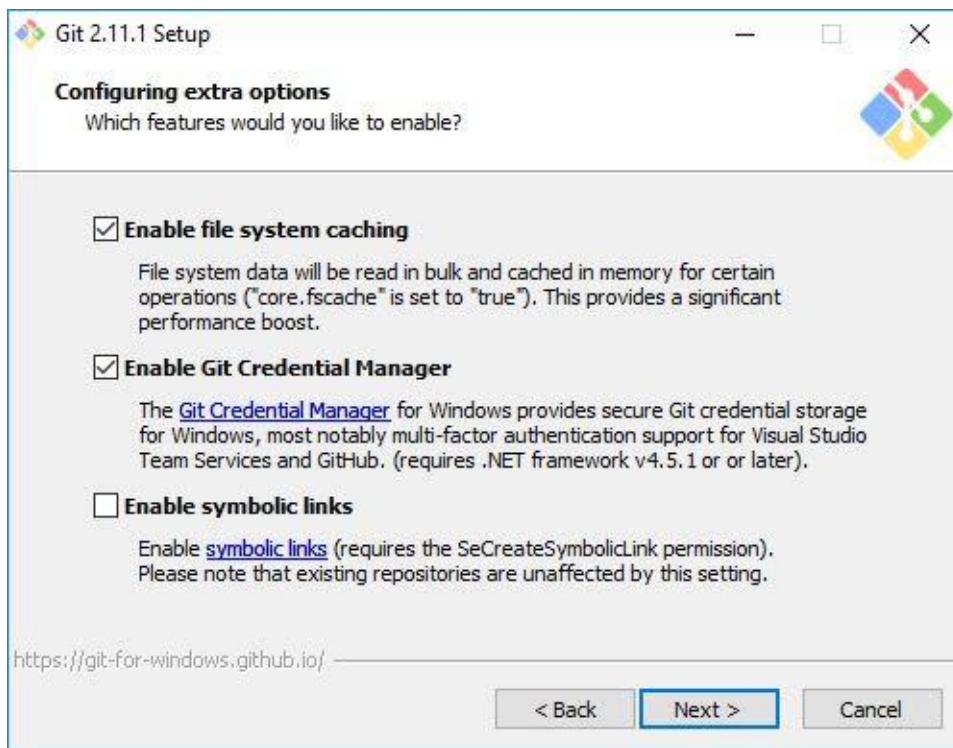
8. Selanjutnya konversi *line ending*. Biarkan saja seperti ini, kemudian klik *Next >*.



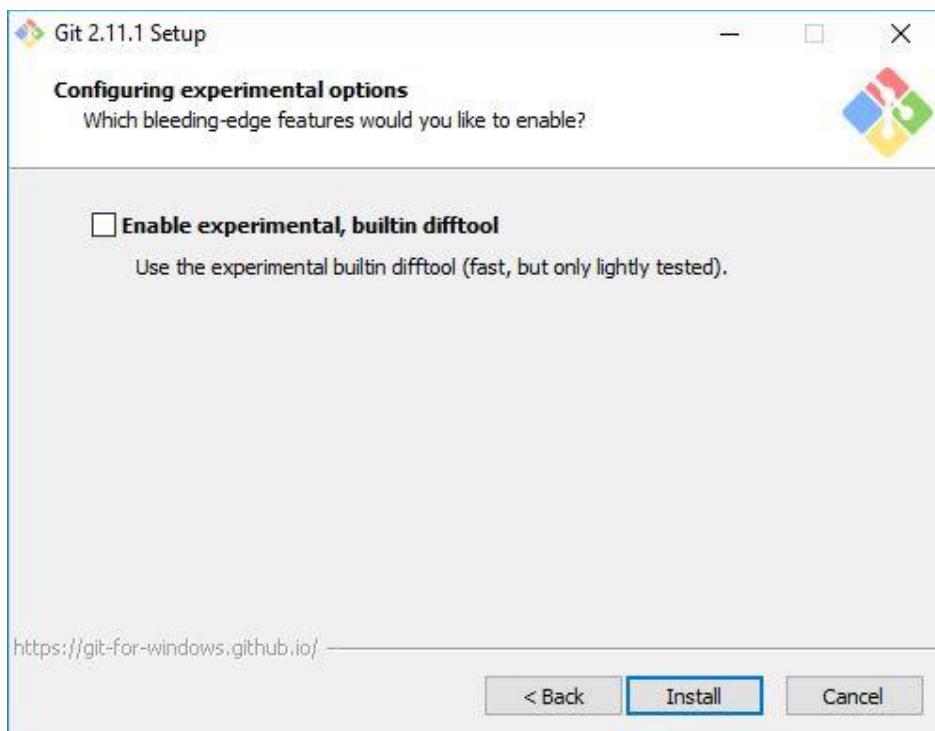
9. Selanjutnya pemilihan emulator terminal. Pilih saja yang bawah, lalu klik *Next >*.



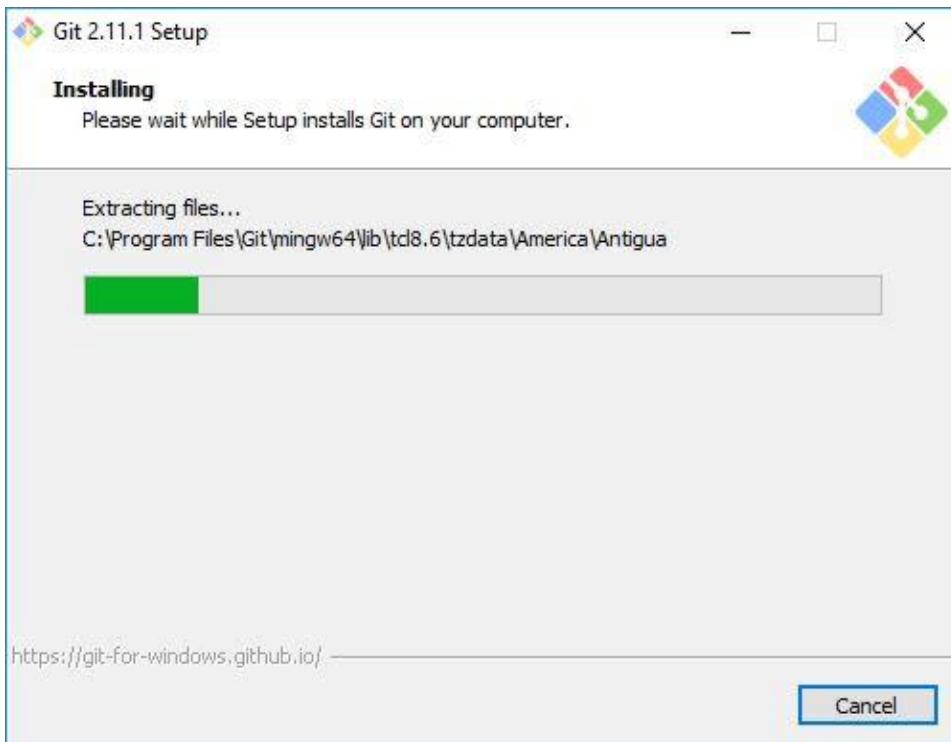
10. Selanjutnya pemilihan opsi ekstra. Klik saja *Next >*.



11. Selanjutnya pemilihan opsi eksperimental, langsung saja klik *Install* untuk instalasi.



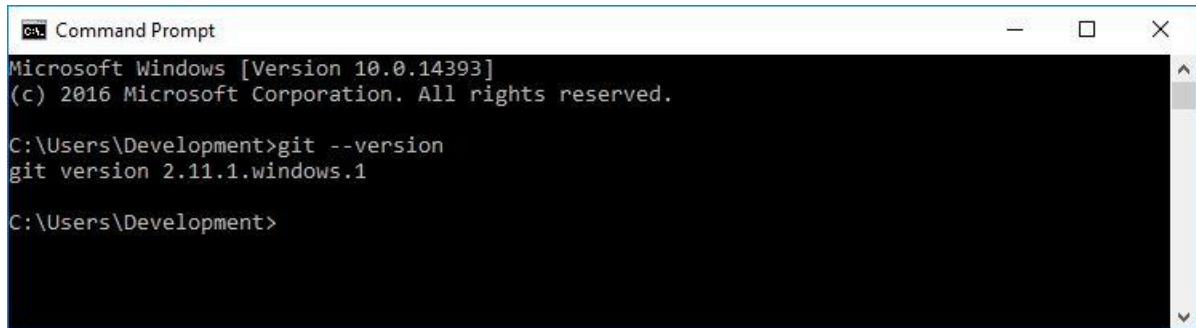
12. Tunggu beberapa saat, instalasi sedang dilakukan.



13. Setelah selesai, kita bisa langsung klik *Finish*.



14. Selamat, Git sudah terinstal di Windows. Untuk mencobanya, silahkan buka CMD atau PowerShell, kemudian ketik perintah git --version.



```
c:\ Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Development>git --version
git version 2.11.1.windows.1

C:\Users\Development>
```

Konfigurasi Awal

Ada beberapa konfigurasi yang harus dipersiapkan sebelum mulai menggunakan Git, seperti *name* dan *email*.

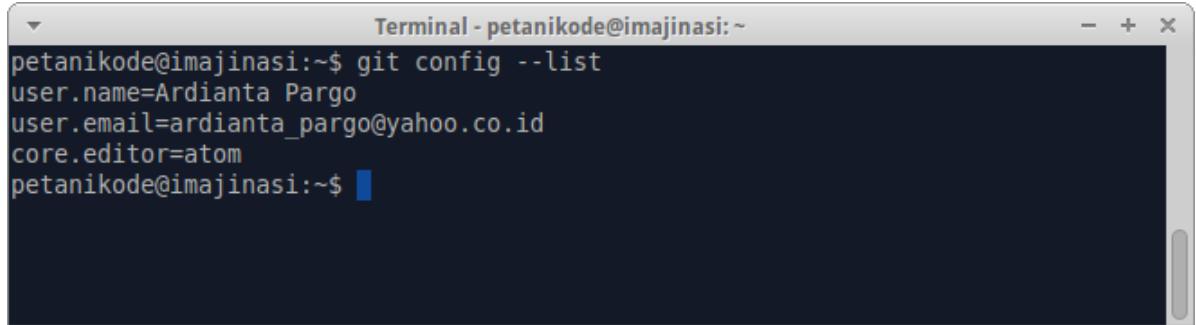
1. Silahkan lakukan konfigurasi dengan perintah berikut ini.

```
git config --global user.name "developer"
git config --global user.email developer@polije.ac.id
```

2. Kemudian periksa konfigurasinya dengan perintah:

```
git config --list
```

Apabila berhasil tampil seperti gambar berikut ini, berarti konfigurasi berhasil.



```
Terminal - petanikode@imajinasi:~
petanikode@imajinasi:~$ git config --list
user.name=Ardianta Pargo
user.email=ardianta_pargo@yahoo.co.id
core.editor=atom
petanikode@imajinasi:~$
```

- Konfigurasi core.editor bersifat opsional, sedangkan *name* dan *email* wajib.

- *Username* dan *email* harus mengikuti akun Github, Gitlab, Bitbucket atau yang lainnya, agar mudah dalam mengintegrasikan.

Revisi Pertama

1. Ubah kondisi tiga file HTML tadi menjadi *staged* dengan perintah git add.

```
git add index.html  
git add about.html  
git add contact.html
```

atau bisa melakukannya seperti ini:

```
git add index.html about.html contact.html
```

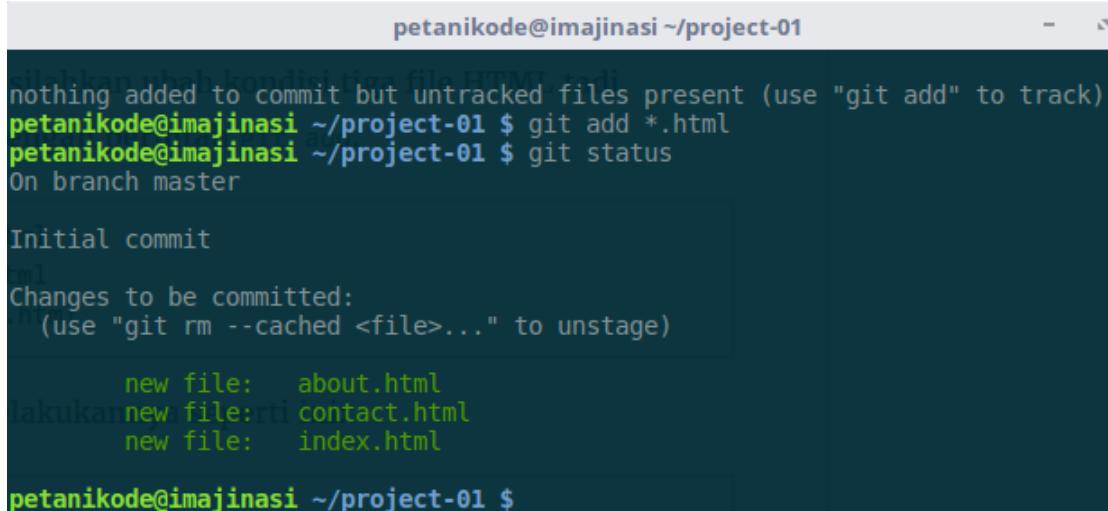
atau:

```
git add *.html
```

atau seperti ini (semua file dan direktori):

```
git add .
```

2. Selanjutnya, cobalah ketik perintah `git status` lagi. Kondisi filenya sekarang akan menjadi *staged*.

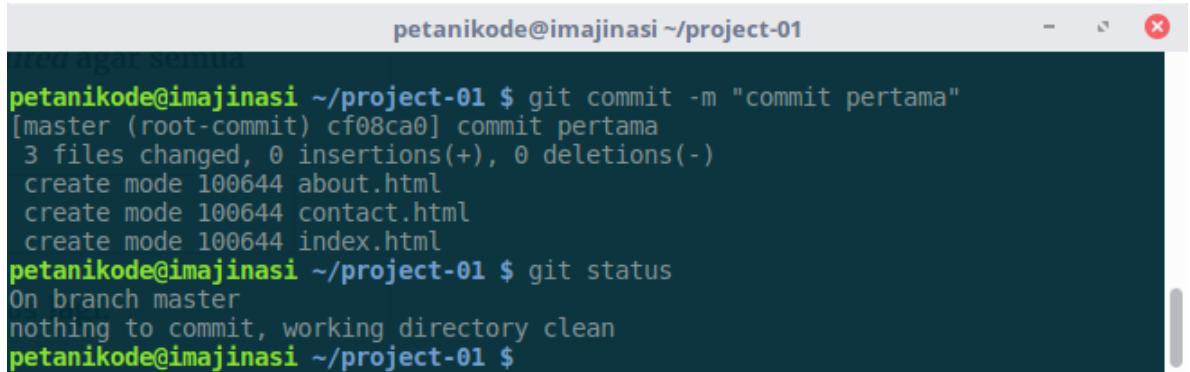


```
petanikode@imajinasi ~/project-01  
  
nothing added to commit but untracked files present (use "git add" to track)  
petanikode@imajinasi ~/project-01 $ git add *.html  
petanikode@imajinasi ~/project-01 $ git status  
On branch master  
  
Initial commit  
  
Changes to be committed:  
(use "git rm --cached <file>..." to unstage)  
  
    new file:   about.html  
lakukan    new file:   contact.html  
    new file:   index.html  
  
petanikode@imajinasi ~/project-01 $
```

3. Lalu, ubah kondisi file tersebut ke *committed* agar semua perubahan disimpan oleh Git.

```
git commit -m "Commit pertama"
```

4. Setelah itu, coba cek dengan perintah git status lagi.



```
petanikode@imajinasi ~/project-01
[...]
petanikode@imajinasi ~/project-01 $ git commit -m "commit pertama"
[master (root-commit) cf08ca0] commit pertama
 3 files changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 about.html
  create mode 100644 contact.html
  create mode 100644 index.html
petanikode@imajinasi ~/project-01 $ git status
On branch master
nothing to commit, working directory clean
petanikode@imajinasi ~/project-01 $
```

5. Selamat, revisi pertama sudah kita buat. Selanjutnya cobalah untuk membuat revisi kedua.

Revisi Kedua

Ceritanya ada perubahan yang akan kita lakukan pada file index.html.

1. Silahkan modifikasi isi file index.html. Sebagai contoh diisikan seperti ini.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Belajar Git - Project 01</title>
  </head>
  <body>
    <p>Hello Semua, Saya sedang belajar Git</p>
  </body>
</html>
```

2. Setelah itu ketik lagi perintah git status.

```
petanikode@imajinasi ~/project-01
nothing to commit, working directory clean
petanikode@imajinasi ~/project-01 $ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
petanikode@imajinasi ~/project-01 $
```

Terlihat di sana, file `index.html` sudah dimodifikasi. Kondisinya sekarang berada dalam *modified*.

3. Lakukan *commit* lagi seperti revisi pertama.

```
git add index.html
git commit -m "ditambahkan isi"
```

- Dengan demikian, revisi kedua sudah disimpan oleh Git.
- Argumen **-m** untuk menambahkan pesan setiap menyimpan revisi.

4. Sekarang Git sudah mencatat dua revisi yang sudah diakukan.

Bisa diibaratkan revisi-revisi ini sebagai *checkpoint* pada Game. Apabila nanti ada kesalahan, kita bisa kembali ke *checkpoint* ini.



f. Hasil dan Pembahasan

1. Dokumentasi tugas berupa laporan secara individu tahap instalasi git yang di buktikan dengan screenshot pada masing-masing PC pada setiap prosesnya.
2. Dokumentasi hasil konfigurasi berupa akun git pribadi.

g. Kesimpulan

Mahasiswa berhasil instalasi Git pada masing-masing PC serta dilanjutkan melakukan konfigurasi dengan menggunakan akun individu serta melakukan revisi pertama dan kedua dengan berhasil.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|--------------------|---|-------|-----|-----|-----|
| 1 | Ketepatan waktu dan ketepatan dalam menjelaskan tahapan instalasi secara berurutan, serta konfigurasi | 1 | 2 | 3 | (4) |
| 2 | Ketepatan waktu dan ketepatan dalam menjelaskan tahapan instalasi, serta konfigurasi | 1 | 2 | (3) | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat dalam menjelaskan tugas, serta konfigurasi | 1 | (2) | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan dalam menjelaskan tugas, serta konfigurasi | (1) | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 3 : PENGENALAN DAN INSTALASI FLUTTER, FLUTTER & DART

| | |
|----------------------------------|---|
| Pokok Bahasan | : Instalasi Flutter |
| Acara Praktikum/Pertemuan | : Minggu 1 / 3 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu melakukan instalasi flutter
2. Mahasiswa menjelaskan maksud setiap langkah & tahap instalasi flutter

b. Indikator

Keberhasilan mahasiswa dalam melakukan revisi pertama pada git, serta memahami setiap command git revisi pertama.

c. Dasar Teori

-

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

Instalasi flutter :

A. Tanpa Android Studio

- SDK Flutter

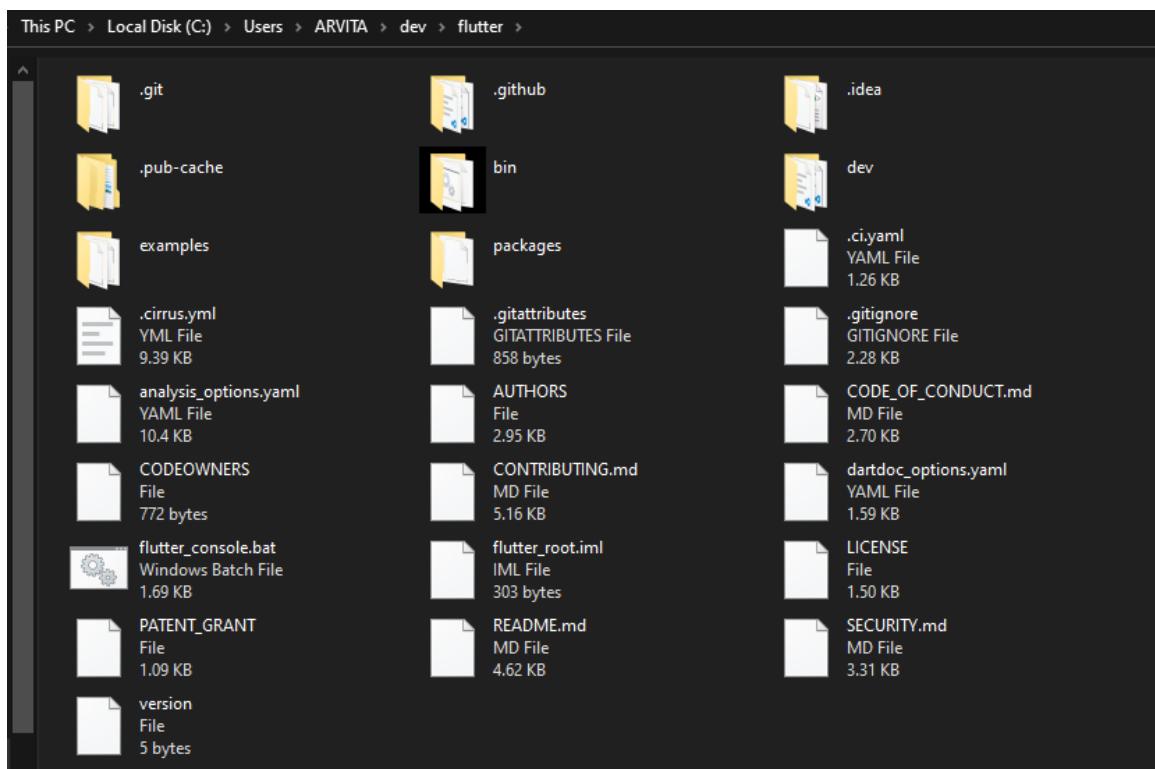
1. Download flutter di <https://flutter.dev/docs/get-started/install>

The screenshot shows the 'Install' section of the Flutter documentation. On the left, there's a sidebar with 'Get started' and '1. Install' selected. Below it is a list of other sections like 'Set up an editor', 'Test drive', etc. A 'From another platform?' section lists various developer environments. The main content area has a title 'Install' and a breadcrumb trail 'Docs > Get started > Install'. It asks to 'Select the operating system on which you are installing Flutter:' with four options: Windows (selected), macOS, Linux, and Chrome OS. A yellow box contains the note: 'Important: If you're in China, first read [Using Flutter in China](#)'. At the bottom right is a link 'Set up an editor'.

2. Pilih sesuai platform laptop, kemudian download.

The screenshot shows the 'Get the Flutter SDK' page for Windows. The sidebar is identical to the previous one. The main content starts with 'Get the Flutter SDK' and step 1: 'Download the following installation bundle to get the latest stable release of the Flutter SDK: [flutter_windows_2.2.3-stable.zip](#)' (the link is highlighted with a red box). Below it is a note: 'For other release channels, and older builds, see the [SDK releases](#) page.' Step 2: 'Extract the zip file and place the contained `flutter` in the desired installation location for the Flutter SDK (for example, `C:\Users\<your-user-name>\Documents`).'. A yellow box contains a warning: '⚠ Warning: Do not install Flutter in a directory like `C:\Program Files\` that requires elevated privileges.' Below that is a code snippet: 'C:\src>git clone https://github.com/flutter/flutter.git -b stable'. At the bottom, it says 'You are now ready to run Flutter commands in the Flutter Console.'

3. Pindahkan hasil download ke dalam folder yang sudah ditentukan, kemudian extract file hasil download.
4. Selanjutkan kita harus setting Environment Variable agar file exe di folder flutter\bin bisa dipanggil dari cmd. Ketik env pada search bar, lalu pilih Edit Environment Variables for Your Account



Best match

Edit environment variables for your account
Control panel

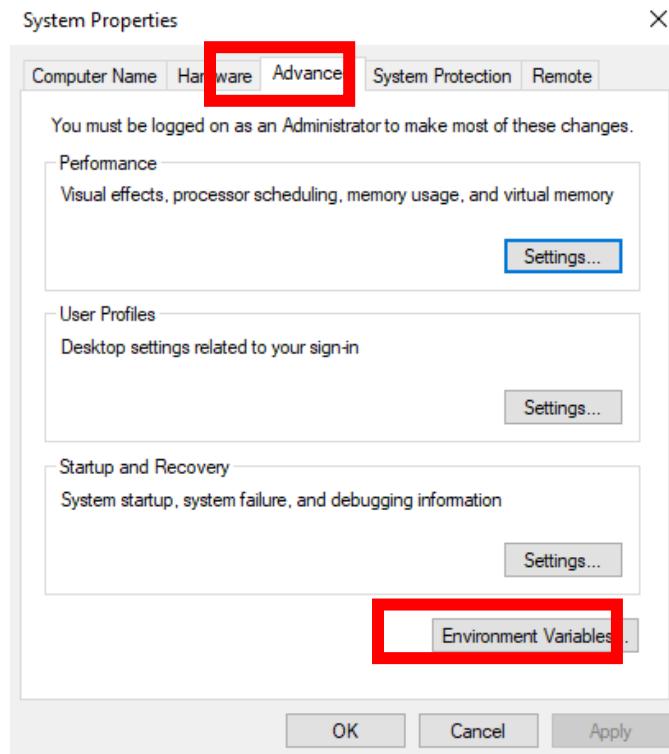
Search the web

- Enviroment Variables - See web results >
- environment variables >
- environment variables os >
- environment variables ruby >
- environment variables windows >
- environment variables & >
- environment variables win7 >
- environment variables js >
- environment variables nx >

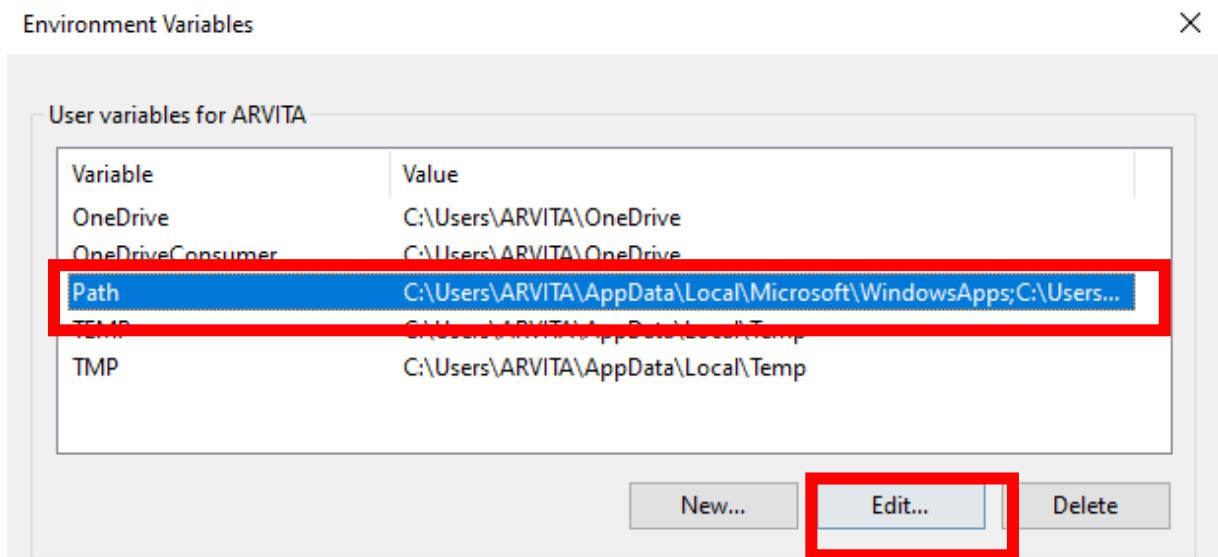
Open

Enviroment Variables

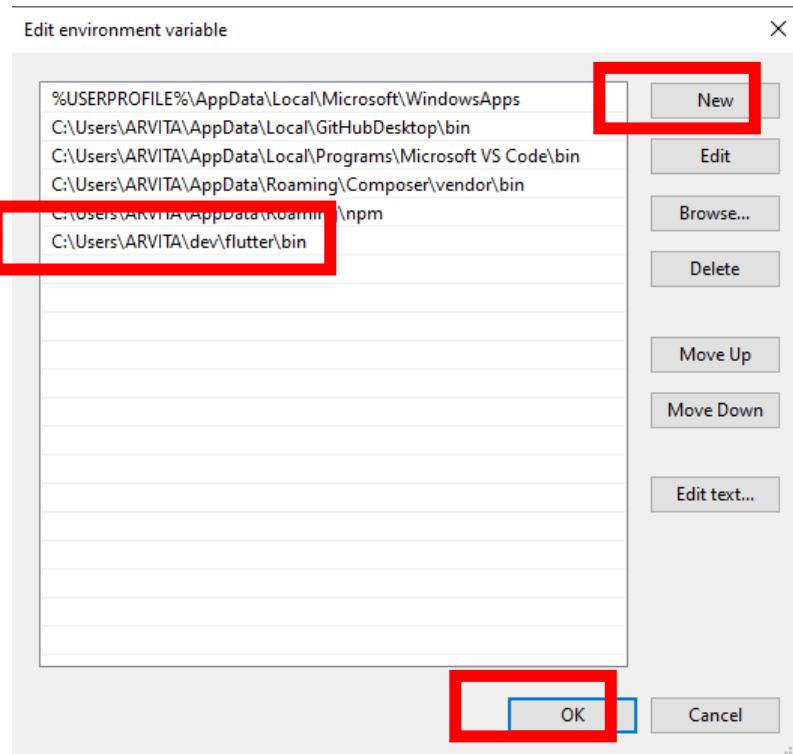
5. pada tab **Advanced**, pilih **Environment Variables**



6. Pada user variables, klik **Path** lalu **Edit**



7. Klik tombol **New**, tambahkan full path C:\Users\ARVITA\dev\flutter\bin, lalu klik **ok**



8. Klik **Ok** pada **Environment Variables**, lalu **close** dan **buka CMD**

9. Jalankan lagi perintah

```
flutter doctor
```

```
Welcome to Flutter! - https://flutter.dev

The Flutter tool uses Google Analytics to anonymously report feature usage
statistics and basic crash reports. This data is used to help improve
Flutter tools over time.

Flutter tool analytics are not sent on the very first run. To disable
reporting, type 'flutter config --no-analytics'. To display the current
setting, type 'flutter config'. If you opt out of analytics, an opt-out
event will be sent, and then no further information will be sent by the
Flutter tool.

By downloading the Flutter SDK, you agree to the Google Terms of Service.
Note: The Google Privacy Policy describes how data is handled in this
service.

Moreover, Flutter includes the Dart SDK, which may send usage metrics and
crash reports to Google.

Read about data we send with crash reports:
https://flutter.dev/docs/reference/crash-reporting

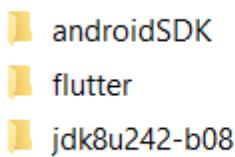
[✓] Flutter (Channel stable, 2.2.3, on Microsoft Windows [Version 10.0.19042.1165], locale en-US)
[✓] Android toolchain - develop for Android devices (Android SDK version 30.0.3)
[✓] Chrome - develop for the web
[✓] Android Studio (version 4.1.0)
[✓] VS Code (version 1.59.1)
[✓] Connected device (2 available)

• No issues found!
```

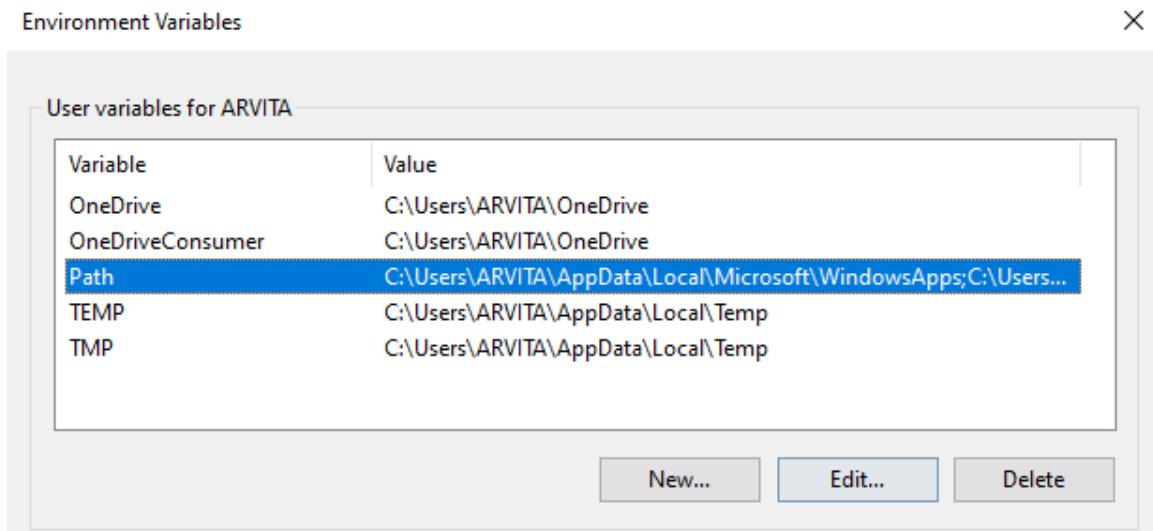
- JDK dan Android SDK

Jika belum menginstal Android JDK atau tidak menginstall Android studio maka lakukan langkah berikut :

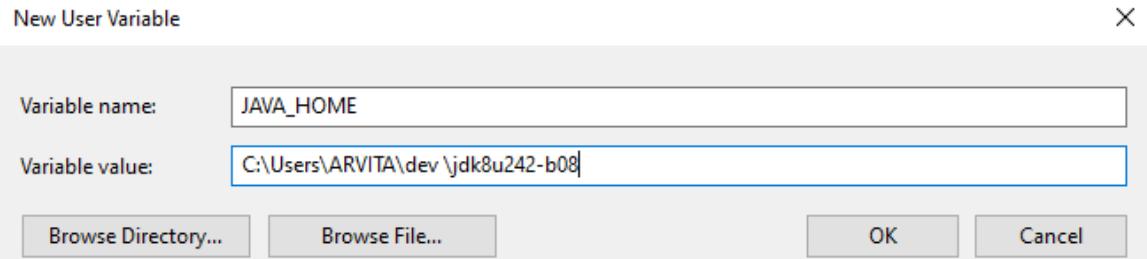
1. Download OPEN JDK di : <https://www.oracle.com/java/technologies/javase-jdk13-downloads.html>. Pilih yang Windows 32 atau Windows 64 bit
2. Ekstrak di folder tadi, sehingga struktur folder sekarang jadi seperti ini



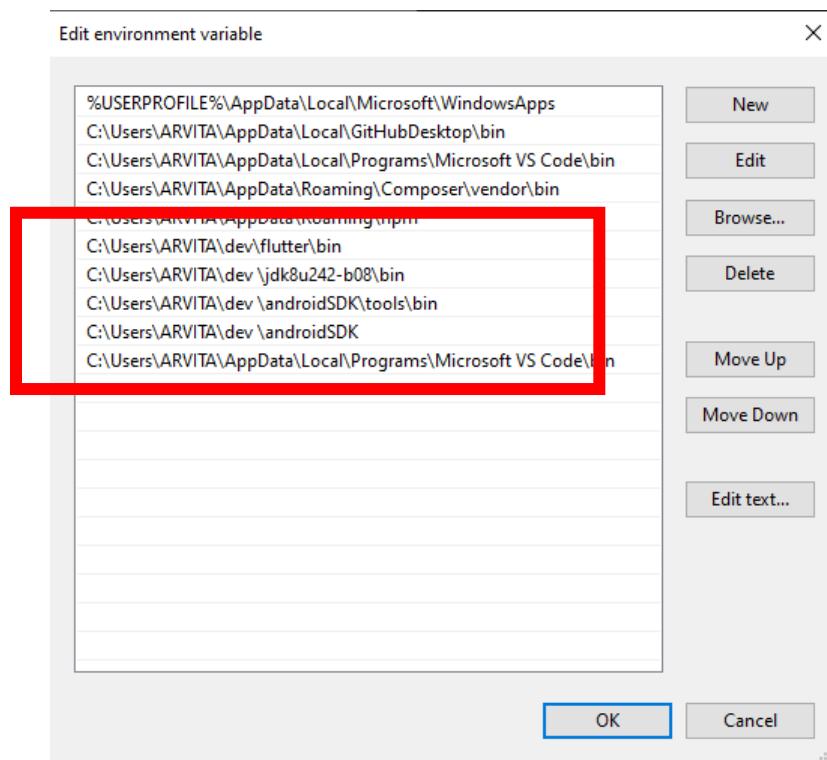
2. Tambahkan Environment Variable untuk JDK. Sesuaikan path JDK. Kalau saya di C:\Users\ARVITA\dev\jdk8u242-b08. Pada Environment Variable, klik New



3. Tambahkan Environment Variable untuk JAVA_HOME, lalu tekan OK, OK juga pada environment variable



4. Silakan download Android SDK di <https://developer.android.com/studio/#command-tools>
5. Saya ekstrak di C:\Users\ARVITA\dev juga agar terkumpul dengan flutter SDK. Nama folder SDK nya silahkan sesuaikan, saya jadikan androidSDK
6. Tambahkan PATH environment variable AndroidSDK\tools\bin



7. Pastikan pada Environment Variable seperti dibawah ini

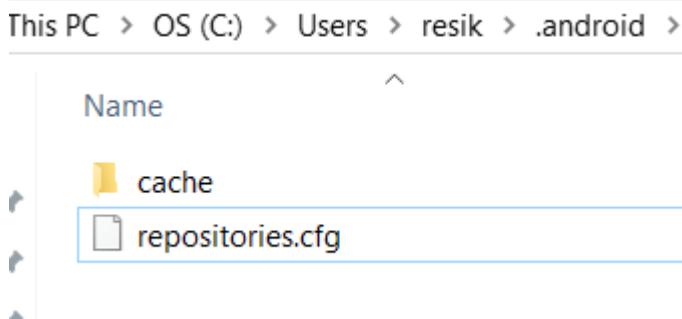
| User variables for ARVITA | |
|---------------------------|---|
| Variable | Value |
| ANDROID_HOME | C:\Users\ARVITA\dev\androidSDK |
| ANDROID_SDK_ROOT | C:\Users\ARVITA\dev\androidSDK\tools |
| JAVA_HOME | C:\Users\ARVITA\dev\jdk8u242-b08 |
| OneDriveConsumer | C:\Users\ARVITA\OneDrive |
| Path | C:\Users\ARVITA\AppData\Local\Microsoft\WindowsApps;C:\Users... |
| TEMP | C:\Users\ARVITA\AppData\Local\Temp |

8. Untuk test konfigurasi, ketik perintah ini pada CMD:

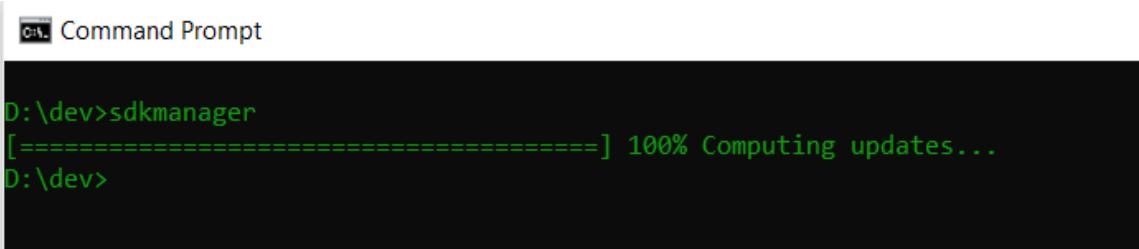
```
sdkmanager
```

9. Jika error seperti itu (repositories.cfg could not be loaded), silakan buat file repositories.cfg folder yang diminta. Contohnya seperti berikut:

```
D:\dev> sdkmanager
Warning: File C:\Users\resik\.android\repositories.cfg could not be loaded.
[=====] 100% Computing updates...
D:\dev>
```



10. Setelah itu ketik ulang sdk manager pada CMD, maka akan muncul seperti ini :



```
D:\dev> sdkmanager
[=====] 100% Computing updates...
D:\dev>
```

11. Tentukan versi Android. Pada CMD ketikkan perintah

```
sdkmanager "build-tools;29.0.3" "platform-tools" "platforms;android-29" "system-images;android-29;default;x86_64"
```

12. Lanjutkan instalasi SDK, tinggal Accept-accept.

```
sdkmanager --update
```

13. Sebelum mulai menggunakan Flutter, kamu harus menyetujui lisensi dari android. Jalankan ini di CMD

```
flutter doctor --android-licenses
```

14. Silakan accept sampai ada tulisan **All SDK package licenses accepted.**

15. Kamu bisa Emulator Android untuk menjalankan aplikasi Androidmu. Sebagai alternatif, bisa gunakan HP atau emulator lain seperti GenyMotion, Bluestacks dan sebagainya.

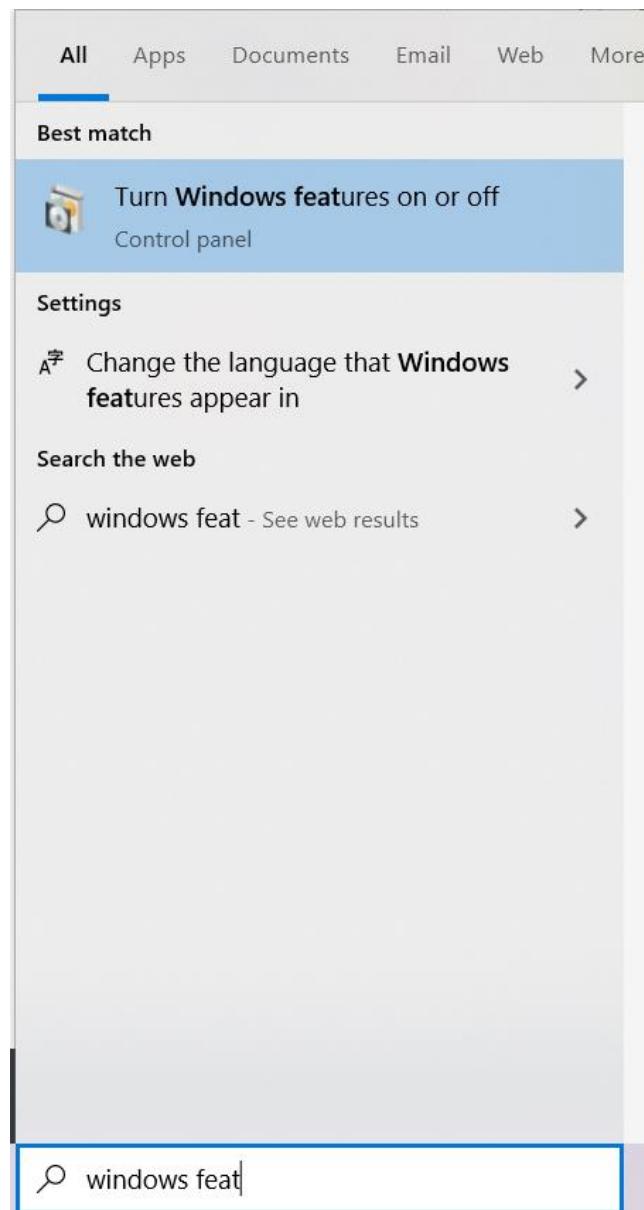
16. silakan install Android Emulator melalui SDK Manager. ketik

```
sdkmanager "emulator"
```

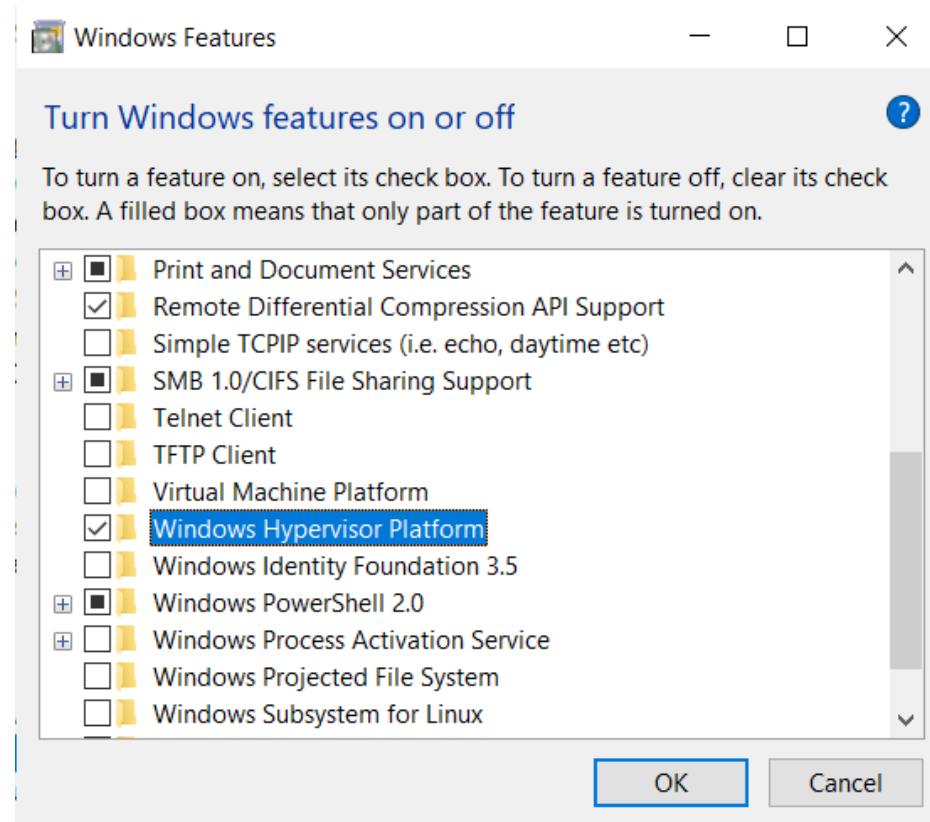
17. Selanjutnya create AVD

```
avdmanager create avd - name Nexus6P - tag 11 - package "system-images;android-29;default;x86_64"
```

18. Aktifkan Windows Hypervisor Platform



19. Aktifkan Windows Hypervisor Platform, lalu OK dan Restart Komputer

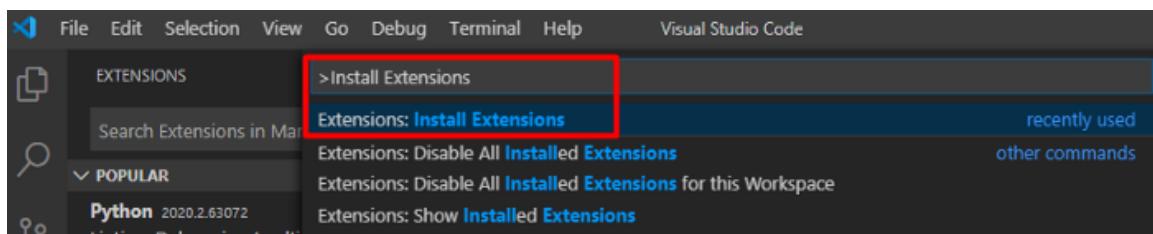


- **Visual Studio Code**

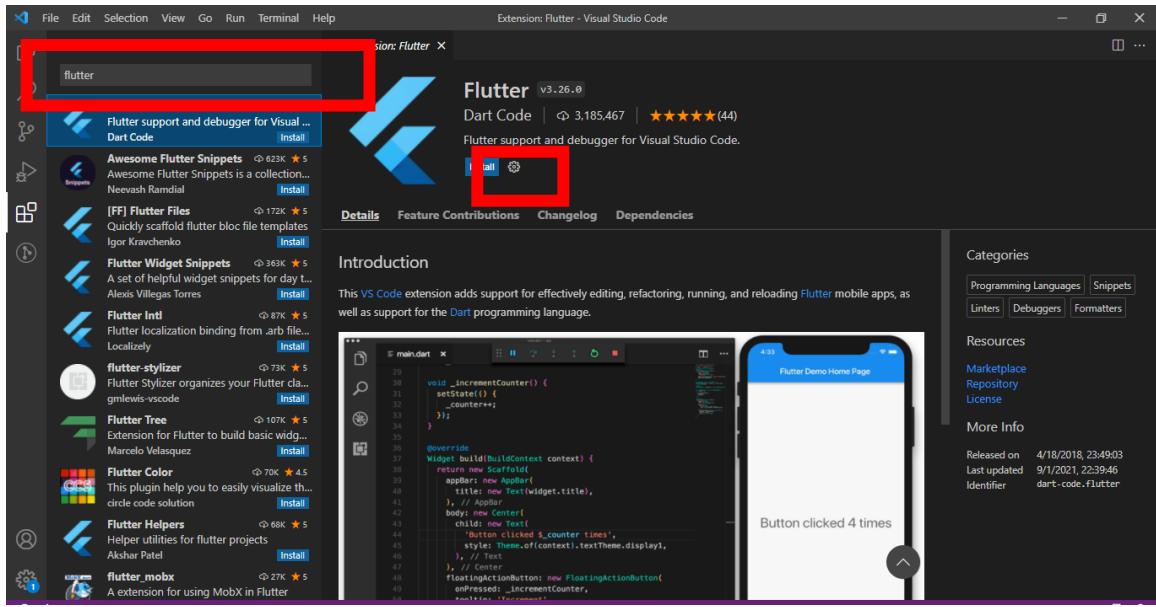
1. Silakan download Visual Studio code di <https://code.visualstudio.com/download>. Lalu lakukan instalasi sampai muncul seperti ini:



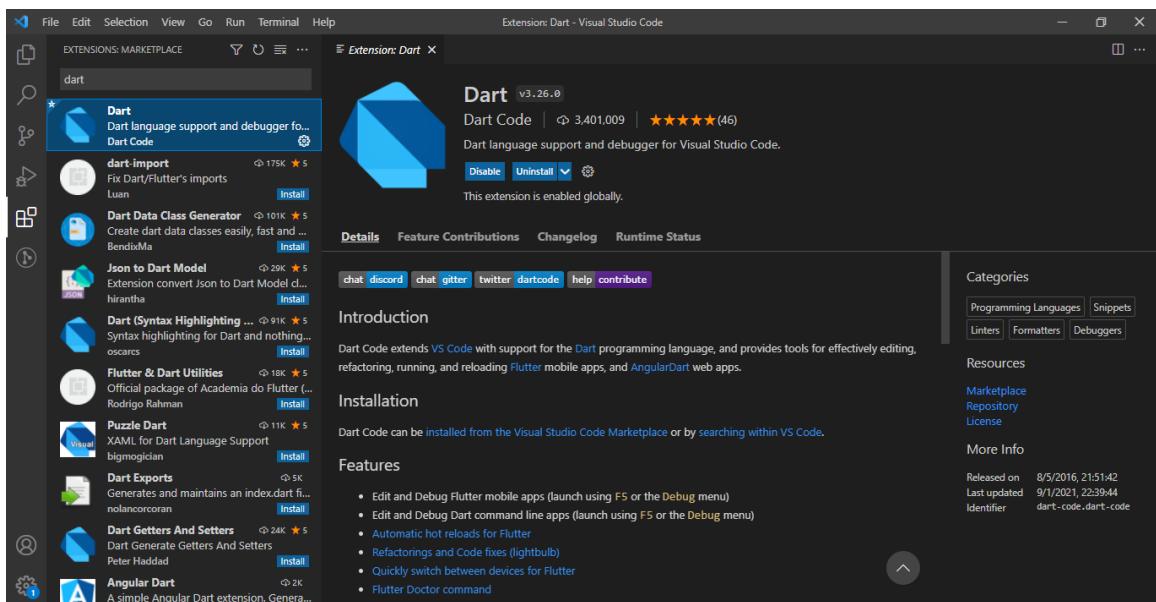
2. Selanjutnya lakukan instalasi extension Flutter dan Dart pada visual studio. Klik View-> Command Pallet -> Ketik Install Extensions lalu enter



3. Search Flutter lalu Install

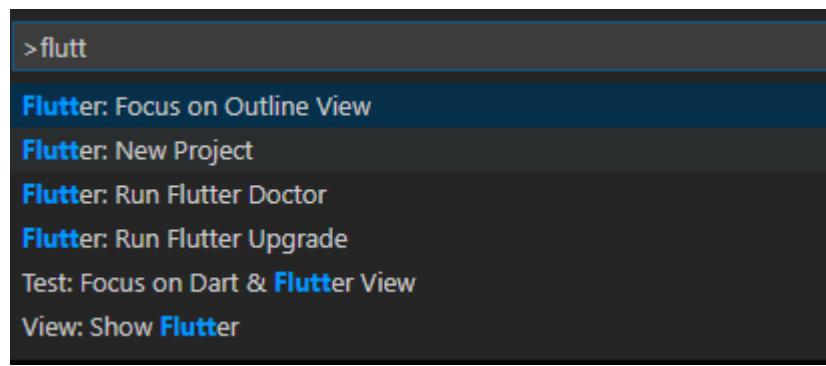


4. Lakukan hal sama untuk extensi Dart (jika belum terinstall)

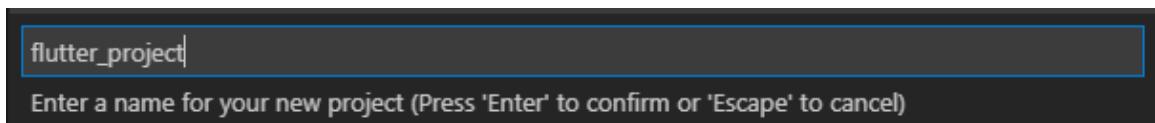


- Uji Coba Project Dengan Flutter

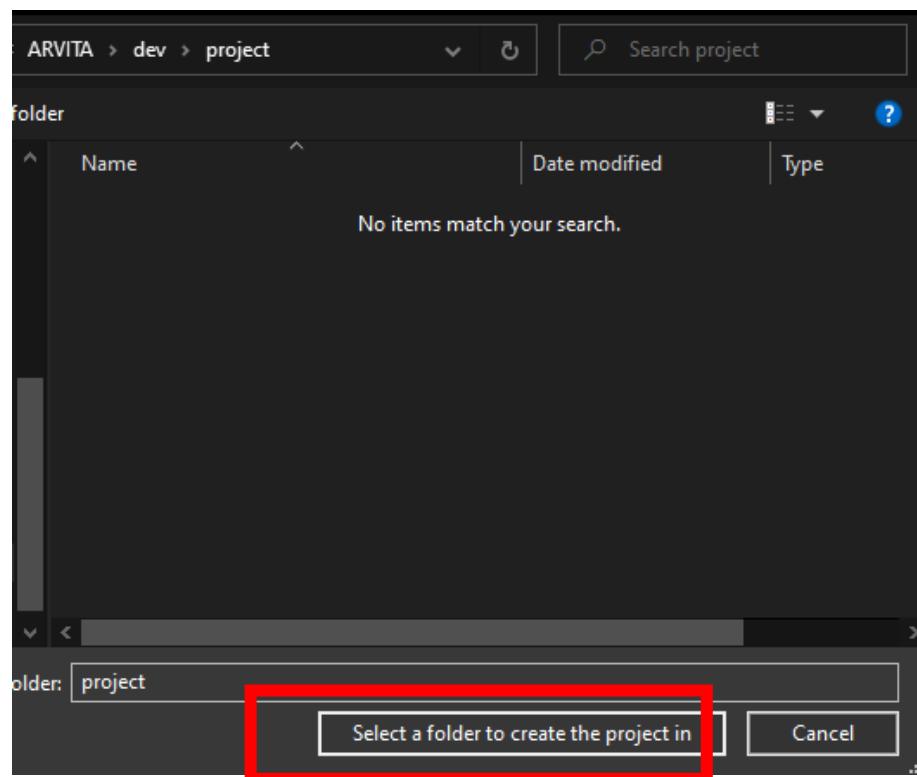
1. Pada Visual Studio Code, tekan F1, lalu ketik Flutter: New Project



2. Contohnya nama projectnya adalah flutter_project



3. Pilih tempat menyimpan, silahkan sesuaikan dengan kebutuhan



```

lib > main.dart
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   // This widget is the root of your application.
9   @override
10  Widget build(BuildContext context) {
11    return MaterialApp(
12      title: 'Flutter Demo',
13      theme: ThemeData(
14        // This is the theme of your application.
15        //
16        // Try running your application with "flutter run". You'll see the
17        // application has a blue toolbar. Then, without quitting the app, try
18        // changing the primarySwatch below to Colors.green and then invoke
19        // "hot reload" (press "r" in the console where you ran "flutter run")
20    );
21  }
22}
23
24
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Wrote 78 files.

All done!

In order to run your application, type:

```
$ cd .
$ flutter run
```

Your application code is in ./lib/main.dart.

exit code 0

Ln 1, Col 1 Spaces:2 UTF-8 CRLF Dart Dart DevTools Flutter:2.2.3 Chrome (web-javascript) Prettier

4. Tekan F5, lalu pilih emulator untuk menjalankan

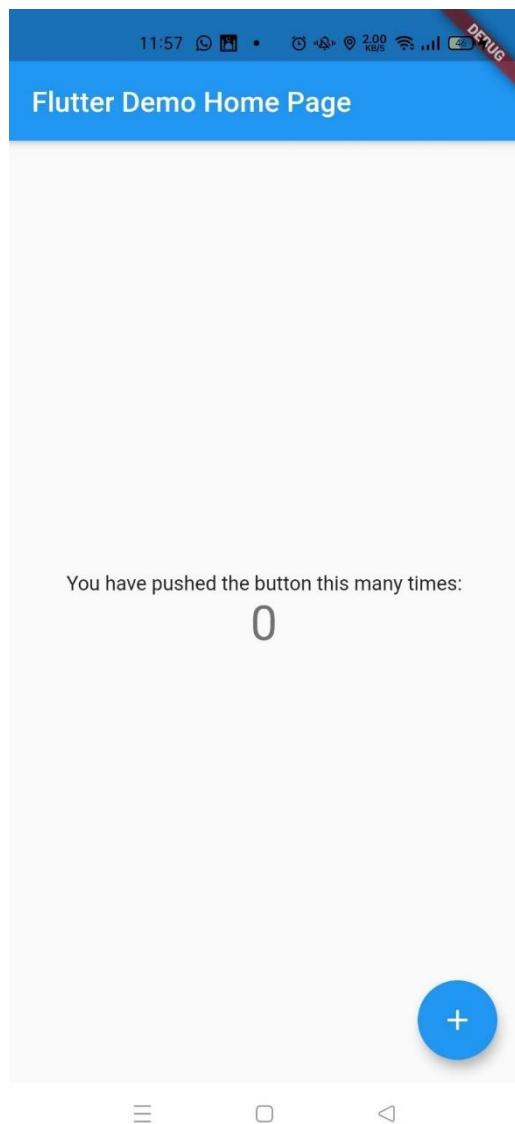
```

RUN AND DEBUG No Configuration ...
VARIABLES
lib > main.dart
24   ), // ThemeData
25   home: MyHomePage(title: 'Flutter Demo Home Page'),
26   ); // MaterialApp
27 }
28 }
29
30 class MyHomePage extends StatefulWidget {
31   MyHomePage({Key? key, required this.title}) : super(key: key);
32
33   // This widget is the home page of your application. It is stateful, meaning
34   // that it has a State object (defined below) that contains fields that affect
35   // how it looks.
36
37   // This class is the configuration for the state. It holds the values (in this
38   // case the title) provided by the parent (in this case the App widget) and
39   // used by the build method of the State. Fields in a Widget subclass are
40   // always marked "final".
41 }

VARIABLES
WATCH
CALL STACK RUNNING PROBLEMS DEBUG CONSOLE TERMINAL Filter (e.g. text, exclude)
0x0, edgeFlags=0x0, pointerCount=1, historySize=0, eventTime=363152065, downTime=363152065, deviceId=4, source=0x1002, disp
layId=0 )
D/ViewRootImpl[MainActivity](23444): processMotionEvent MotionEvent { action=ACTION_CANCEL, actionButton=0, id[0]=0, x[0]=5
29.0, y[0]=532.0, toolType[0]=TOOL_TYPE_FINGER, id[1]=1, x[1]=789.0, y[1]=579.0, toolType[1]=TOOL_TYPE_FINGER, id[2]=2, x
[2]=288.0, y[2]=722.0, toolType[2]=TOOL_TYPE_FINGER, buttonState=0, classification=NONE, metaState=0, flags=0x0, edgeFlags=
0x0, pointerCount=3, historySize=0, eventTime=363152108, downTime=363152065, deviceId=4, source=0x1002, displayId=0 }
D/ViewRootImpl[MainActivity](23444): dispatchPointerEvent handled=true, event=MotionEvent { action=ACTION_CANCEL, actionBut
ton=0, id[0]=0, x[0]=529.0, y[0]=532.0, toolType[0]=TOOL_TYPE_FINGER, id[1]=1, x[1]=789.0, y[1]=579.0, toolType[1]=TOOL_TY
P_E_FINGER, id[2]=2, x[2]=288.0, y[2]=722.0, toolType[2]=TOOL_TYPE_FINGER, buttonState=0, classification=NONE, metaState=0, f
lags=0x0, edgeFlags=0x0, pointerCount=3, historySize=0, eventTime=363152188, downTime=363152065, deviceId=4, source=0x1002,
displayId=0 }

BREAKPOINTS
■ All Exceptions
✓ Uncaught Exceptions

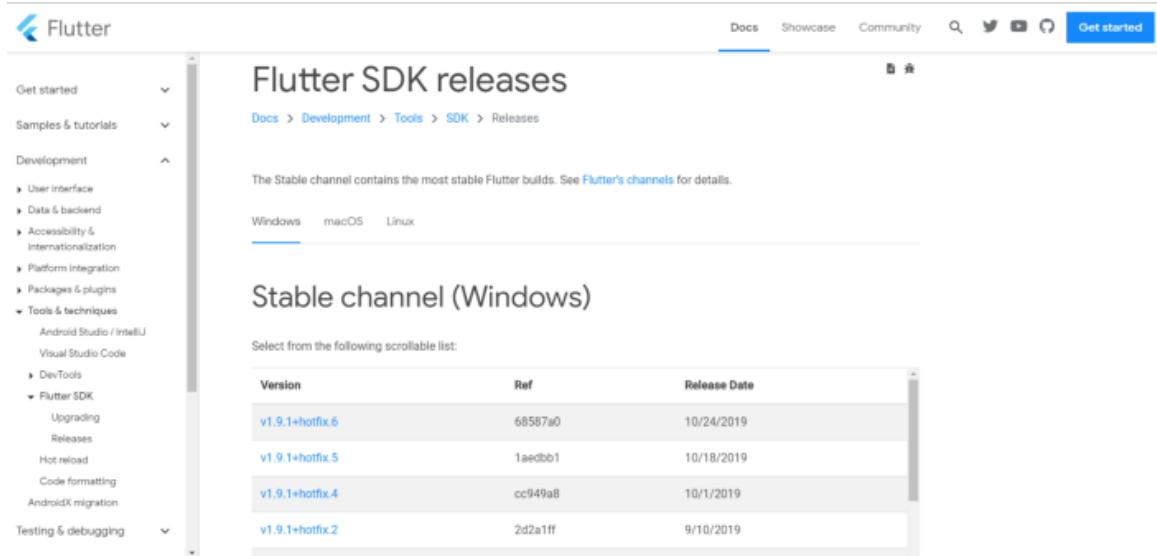
```



B. Menggunakan Android Studio

1. Unduh Flutter SDK <https://flutter.dev/docs/development/tools/sdk/releases>

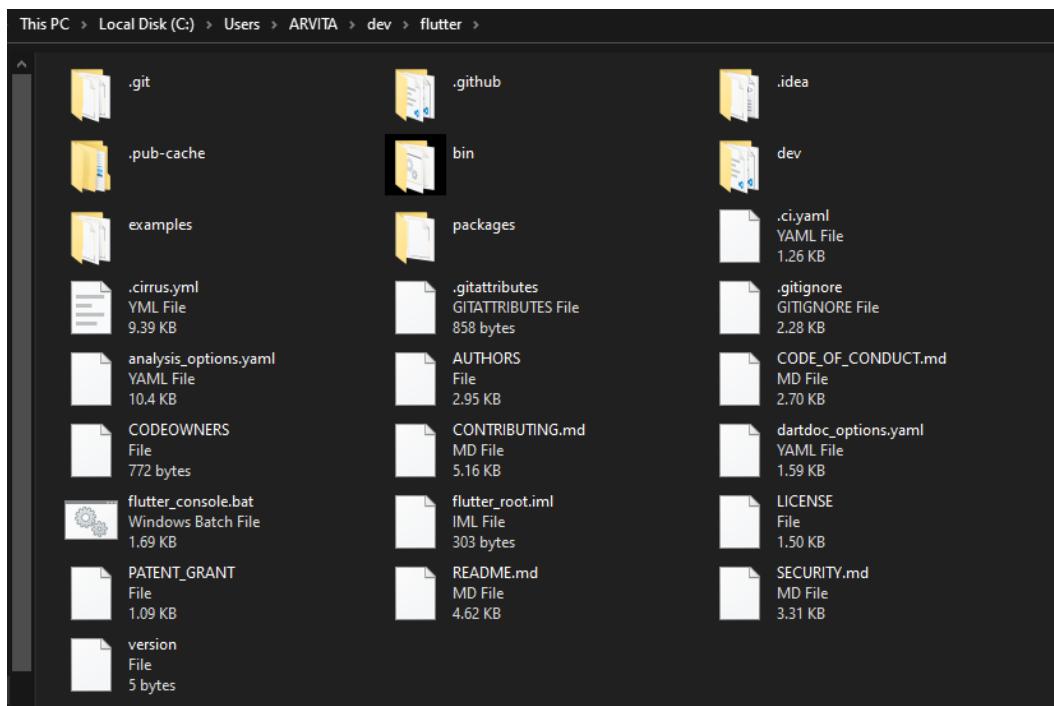
Untuk pemula, untuk memilih Flutter SDK versi terbaru (paling atas) dengan tipe **Stable Channel**.



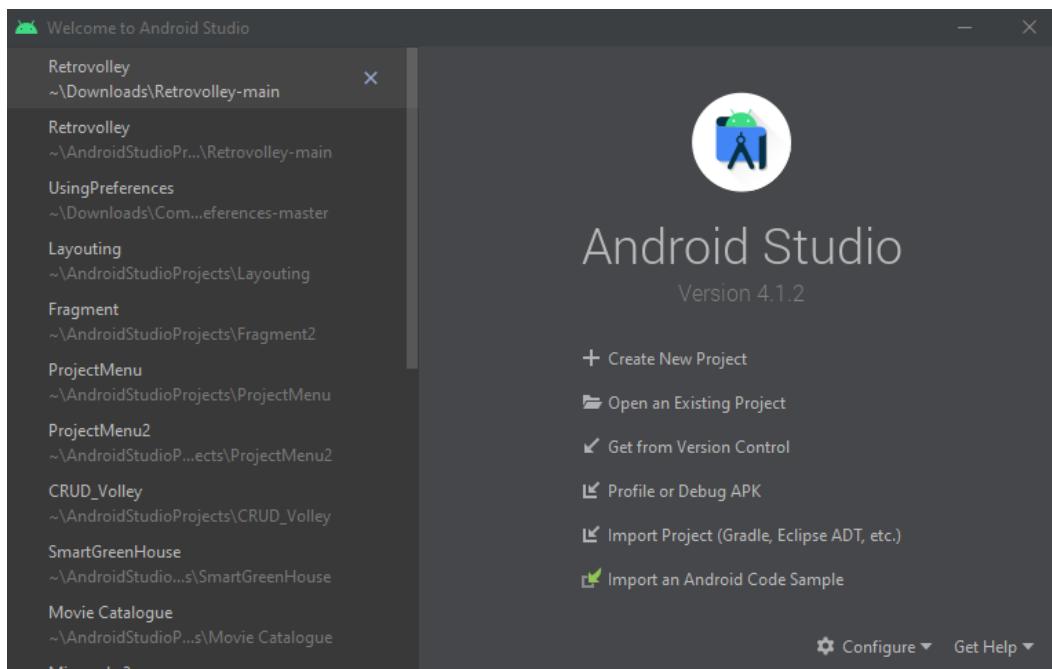
The screenshot shows the 'Flutter SDK releases' page. The left sidebar has sections like 'Get started', 'Samples & tutorials', 'Development' (expanded to show 'User interface', 'Data & backend', etc.), 'Tools & techniques' (expanded to show 'Android Studio / IntelliJ', 'Visual Studio Code', 'DevTools', 'Flutter SDK' (selected), 'Upgrading', 'Releases', 'Hot reload', 'Code formatting', 'AndroidX migration', and 'Testing & debugging'). The main content area is titled 'Flutter SDK releases' and shows the 'Stable channel (Windows)' section. It says 'The Stable channel contains the most stable Flutter builds. See [Flutter's channels](#) for details.' Below this are tabs for 'Windows' (selected), 'macOS', and 'Linux'. A table lists releases:

| Version | Ref | Release Date |
|-----------------|---------|--------------|
| v1.9.1+hotfix.6 | 68587a0 | 10/24/2019 |
| v1.9.1+hotfix.5 | 1aedbb1 | 10/18/2019 |
| v1.9.1+hotfix.4 | cc949a8 | 10/1/2019 |
| v1.9.1+hotfix.2 | 2d2a1ff | 9/10/2019 |

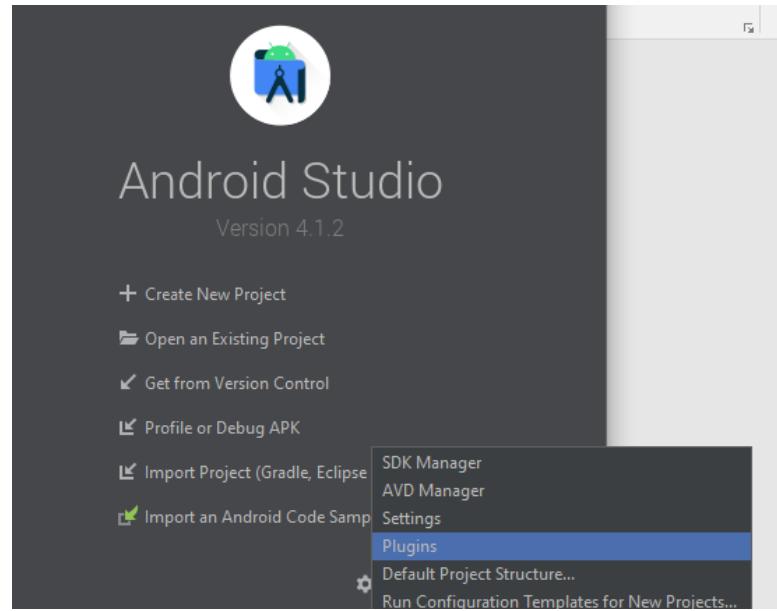
2. Ekstrak berkas ZIP/RAR Flutter SDK yang sebelumnya telah anda unduh, ke dalam direktori folder yang anda inginkan.
3. **Simpanlah di dalam direktori yang mudah diakses.** Contoh: C:\Users\ARVITA\dev\flutter



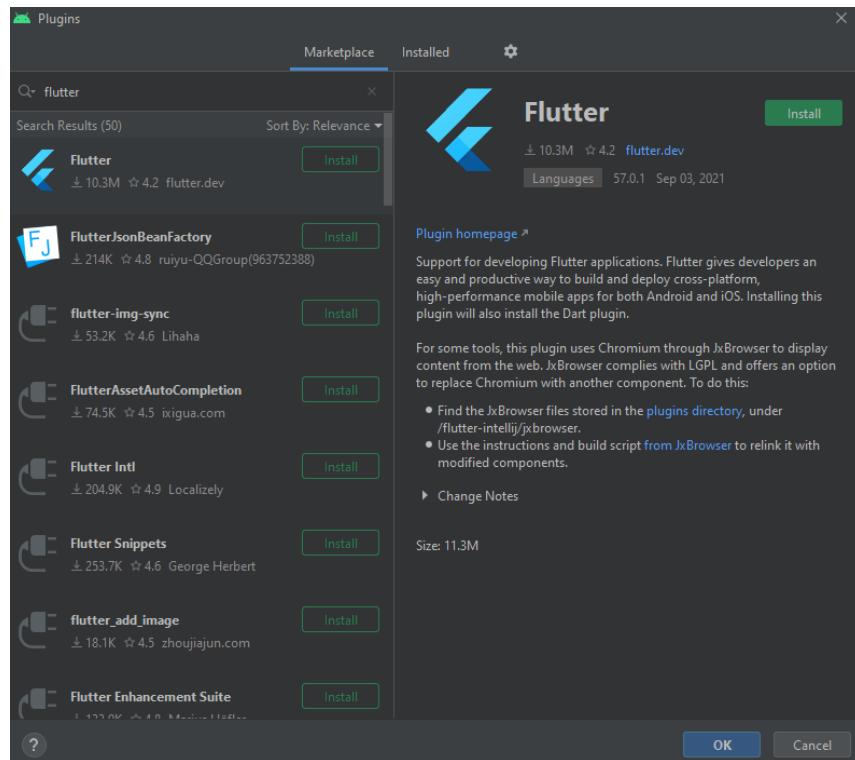
4. Buka *Android Studio* yang telah terpasang pada komputer anda, lalu tunggu hingga halaman awal *Android Studio* ditampilkan.



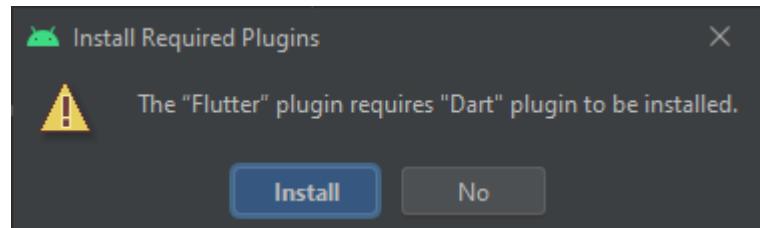
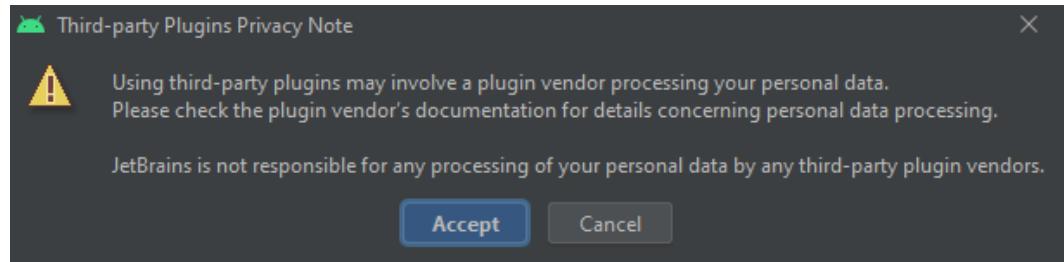
5. Buka *Configure -> Plugins*



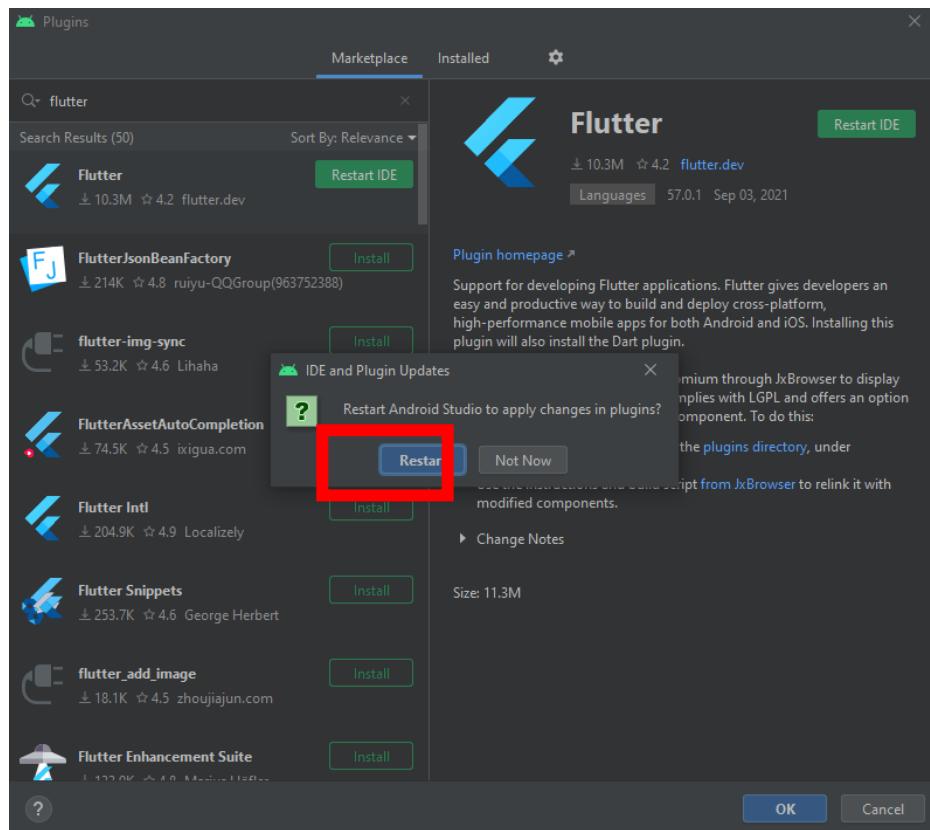
- Setelah muncul jendela **Plugins**, maka pilihlah tab **Marketplace**, kemudian lakukan pencarian dengan kata kunci: “**flutter**”. Terakhir tekan tombol **Install**



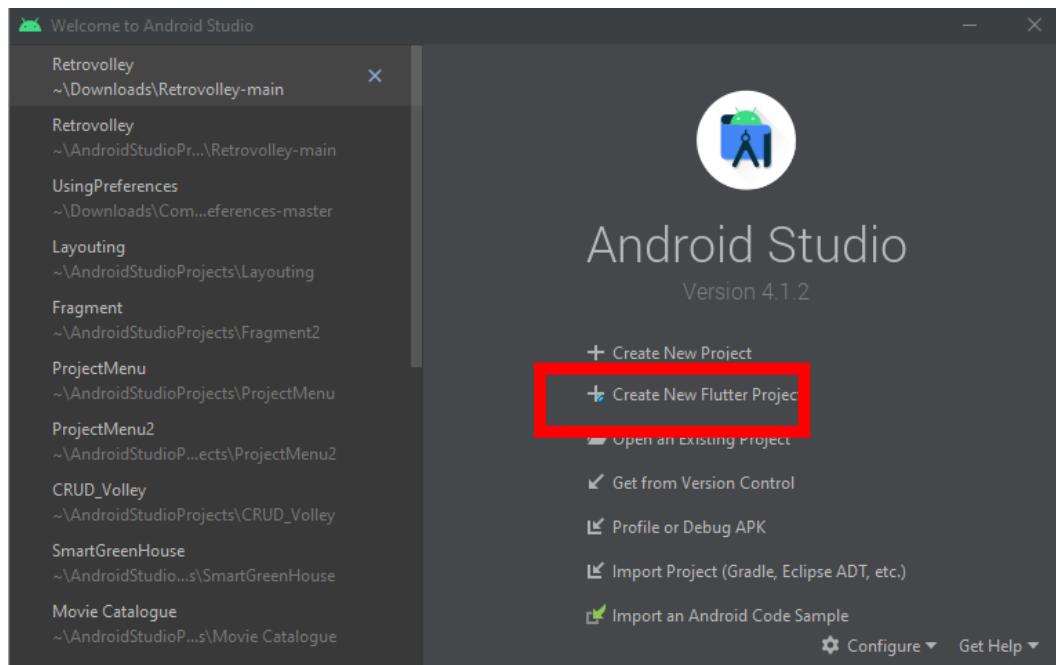
- Jika anda mendapat *notification* berikut ini, pilih saja “Yes”



8. Tunggu proses pemasangan *plugin* hingga selesai.
9. Setelah itu pilihlah tombol “**Restart IDE**”.

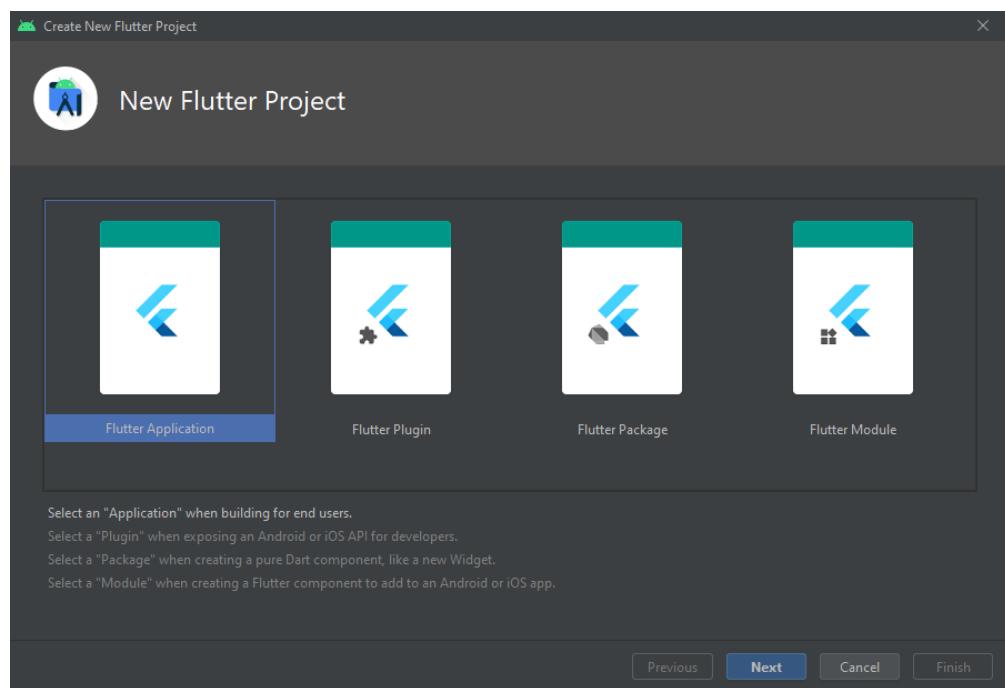


10. Tunggu proses mulai ulang *Android Studio*, hingga muncul kembali halaman awalnya seperti ini, dan lihatlah perbedaannya.

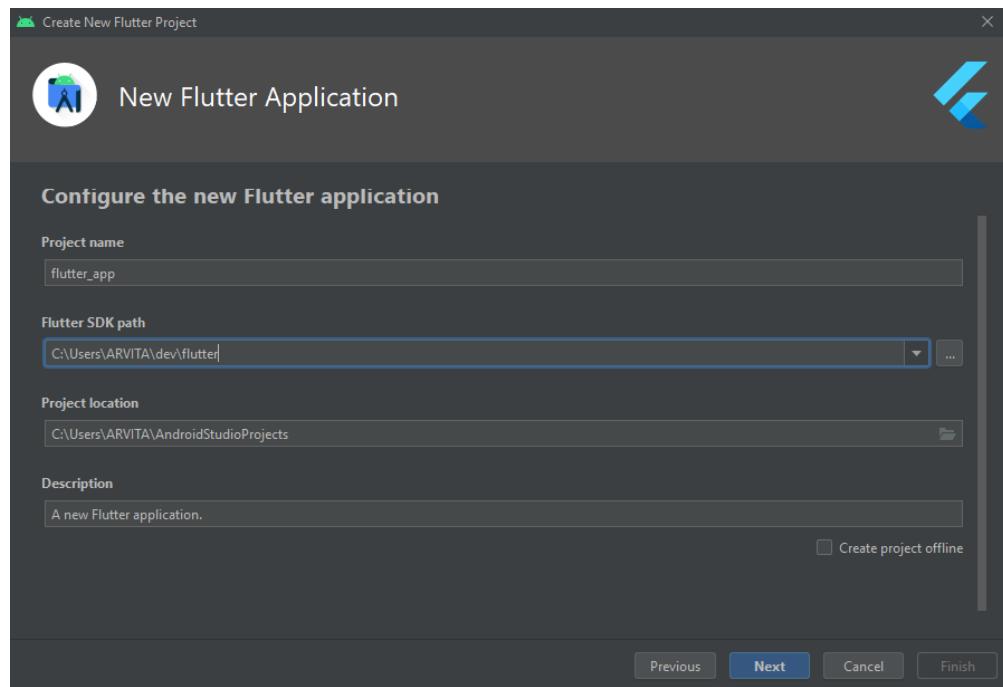


11. Pilihlah “*Create new flutter project*” di halaman awal *Android Studio* anda.

12. Kemudian, pilihlah “*Flutter Application*” lalu klik “*Next*”.



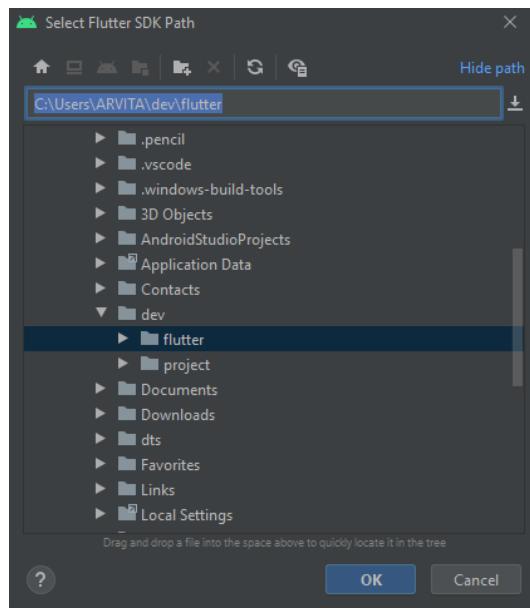
13. lalu akan muncul jendela berikut ini:



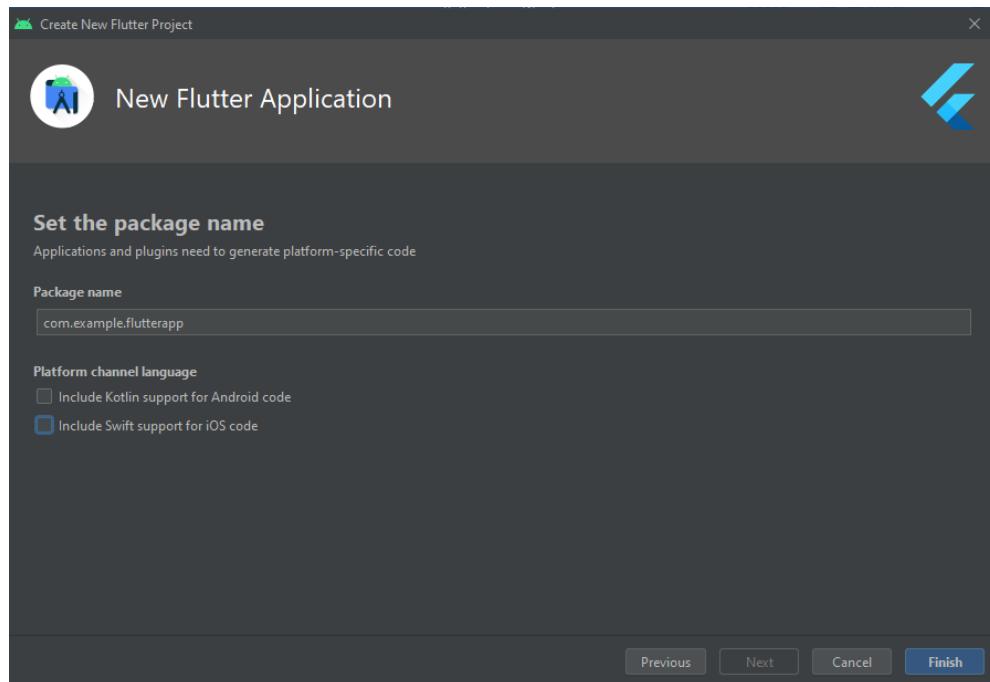
14. Silahkan beri *project name* sesuai dengan keinginan anda. Lakukan penamaan dengan menggunakan **huruf kecil dan tanpa spasi**.

15. Kemudian pada bagian *flutter SDK path*, tentukanlah direktori penyimpanan folder Flutter SDK yang telah kita unduh pada **langkah no. 1–2** di atas.

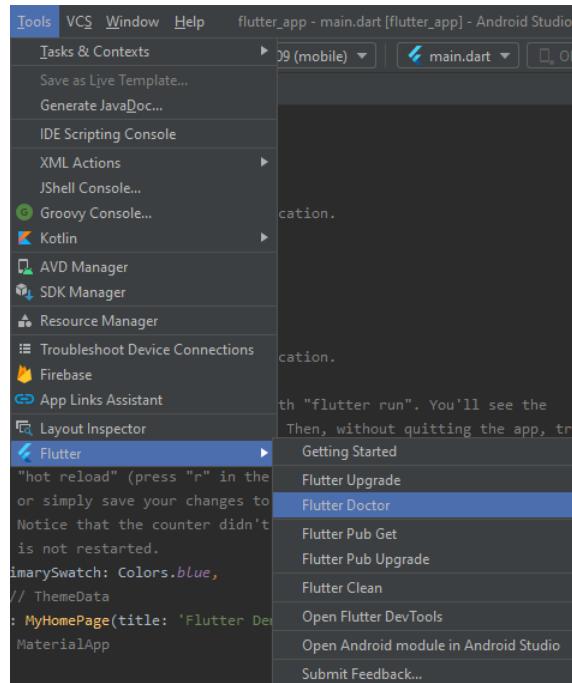
16. Tekan tombol diatas untuk menentukan direktori Flutter SDK



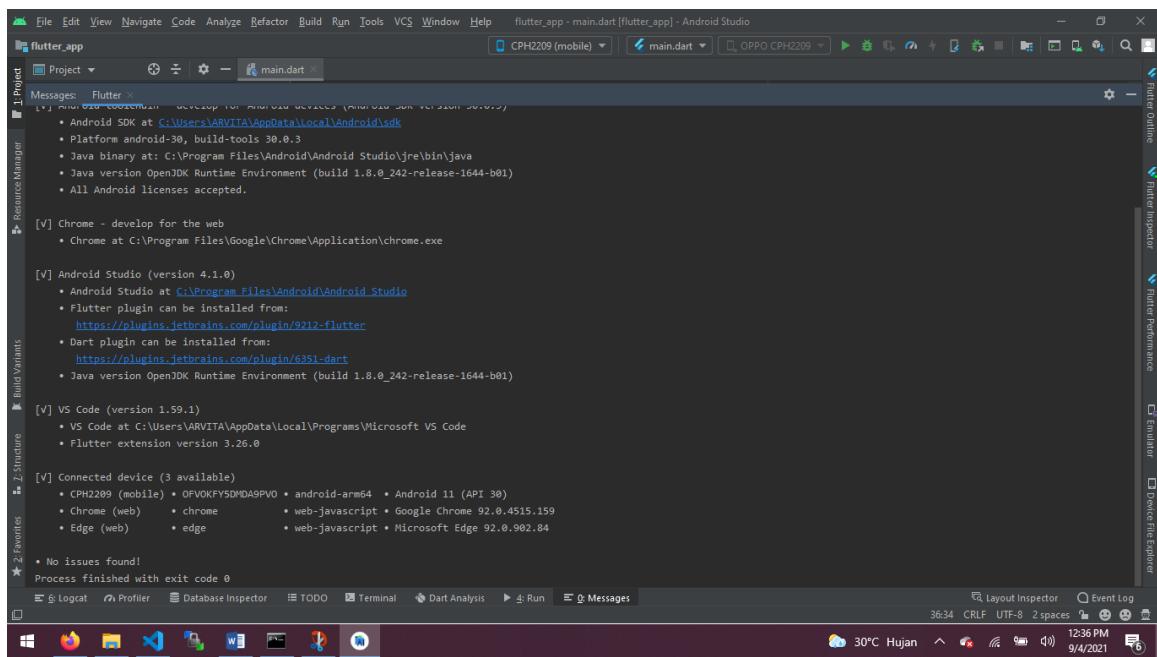
17. Untuk bagian *project location*, silahkan anda tentukan tempat untuk menyimpan project anda.
Direkomendasikan agar nama folder tidak mengandung spasi.
18. Jika semua konfigurasi telah selesai, silahkan klik “*Next*”.
19. Kemudian pada jendela selanjutnya, anda harus menentukan domain aplikasi, untuk keperluan penamaan direktori *package* pada aplikasi anda. Untuk bagian *AndroidX* sebaiknya **di ceklis** dan untuk ***Platform channel language* tidak perlu di ceklis**.



20. Jika telah selesai semua, klik “*Finish*”.
21. Tunggu proses *building* hingga selesai.
22. Project telah berhasil dibuat
23. Di dalam Android Studio, bukalah menu **Tools -> Flutter -> Flutter Doctor** seperti contoh gambar berikut:



24. Lihatlah hasil keterangan yang muncul dalam tab **Messages**, seperti gambar berikut ini:



25. Jika ada issue seperti berikut :

```
Messages: Flutter < D:\Android\flutter\bin\flutter.bat doctor --verbose
[✓] Flutter (Channel stable, v1.9.1+hotfix.6, on Microsoft Windows [Version 10.0.18362.418], locale en-ID)
  • Flutter version 1.9.1+hotfix.6 at D:\Android\flutter
  • Framework revision 68587a0916 (8 weeks ago), 2019-09-13 19:46:58 -0700
  • Engine revision b863200c37
  • Dart version 2.5.0

[!] Android toolchain - develop for Android devices (Android SDK version 29.0.2)
  • Android SDK at D:\Android\sdk
  • Android NDK location not configured (optional; useful for native profiling support)
  • Platform android-29, build-tools 29.0.2
  • ANDROID_HOME = D:\Android\sdk
  • Java binary at: D:\Android\Android Studio\jre\bin\java
  • Java version OpenJDK Runtime Environment (build 1.8.0_202-release-1483-b03)
! Some Android licenses not accepted. To resolve this, run: flutter doctor --android-licenses

[✓] Android Studio (version 3.5)
  • Android Studio at D:\Android\Android Studio
  • Flutter plugin version 41.0.2
  • Dart plugin version 191.8593
  • Java version OpenJDK Runtime Environment (build 1.8.0_202-release-1483-b03)

[!] Connected device
! No devices available

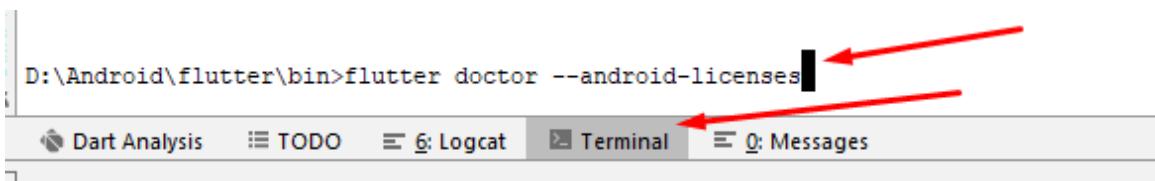
! Doctor found issues in 2 categories.
Process finished with exit code 0
```

Penyebab *issues* tersebut adalah:

Some Android licenses not accepted.... : diharuskan menyetujui lisensi (syarat & ketentuan) dari Android terlebih dahulu.

No devices available : Tidak ada perangkat ponsel Android yang terhubung ke komputer.

26. Untuk mengatasi *issues* tersebut, bukalah tab **Terminal**, kemudian masuklah ke folder “**bin**” dalam **direktori penyimpanan Flutter SDK** anda pada tahap nomor 1–2.



27. Setelah masuk folder “**bin**”, ketiklah perintah android-licenses seperti pada contoh dalam gambar diatas!

28. Berikan **input: y** untuk menyetujui license

The screenshot shows the Android Studio interface with the 'Terminal' tab selected. The terminal window displays the command `D:\Android\flutter\bin>flutter doctor --android-licenses`. A red box highlights the prompt `Review license that has not been accepted (y/N)? y`. The user inputs `y`, which is also highlighted with a red box. The terminal then shows the acceptance of all SDK package licenses.

```
D:\Android\flutter\bin>flutter doctor --android-licenses
Warning: File C:/Users/IIUNIDA/.android/repositories.cfg could not be loaded.

1 of 7 SDk package license not accepted.] 100% Computing updates...
Review license that has not been accepted (y/N)? y
input: y

1/l: License android-sdk-arm-dbt-licenses:
-----
Terms and Conditions

This is the Android Software Development Kit License Agreement

1. Introduction

January 16, 2019      input: y
-----
Accept? (y/N): y
All SDK package licenses accepted
```

29. Kemudian, ketik perintah **flutter doctor** lagi untuk mengecek apakah *issues* sudah diperbaiki atau belum.

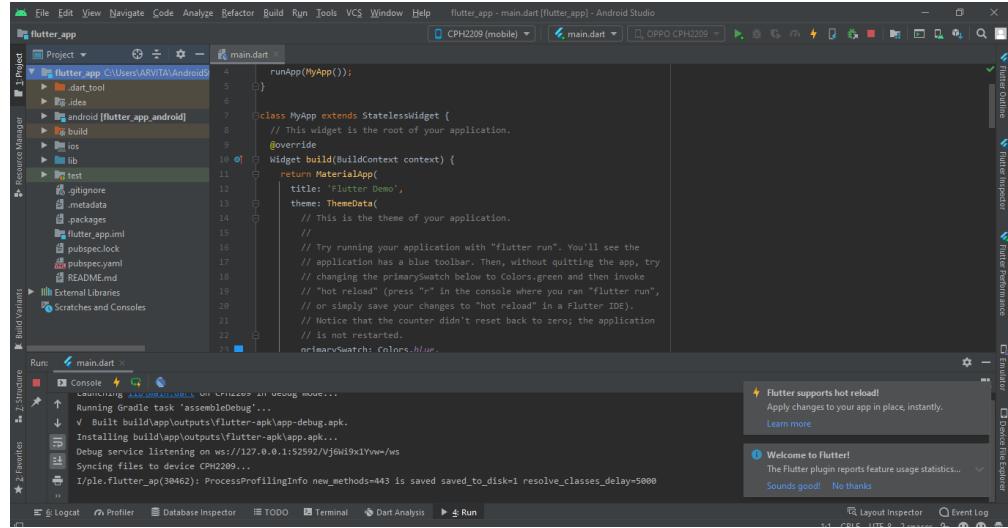
The screenshot shows the Android Studio interface with the 'Terminal' tab selected. The terminal window displays the command `D:\Android\flutter\bin>flutter doctor`. A red arrow points to this command. The output shows a doctor summary with one issue: `No devices available`. The terminal also indicates that the doctor found issues in one category.

```
D:\Android\flutter\bin>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, v1.9.1+hotfix.6, on Microsoft Windows [Version 10.0.18362.418], locale en-ID)
[✓] Android toolchain - develop for Android devices (Android SDK version 29.0.2)
[✓] Android Studio (version 3.5)
[!] Connected device
    ! No devices available

! Doctor found issues in 1 category.
```

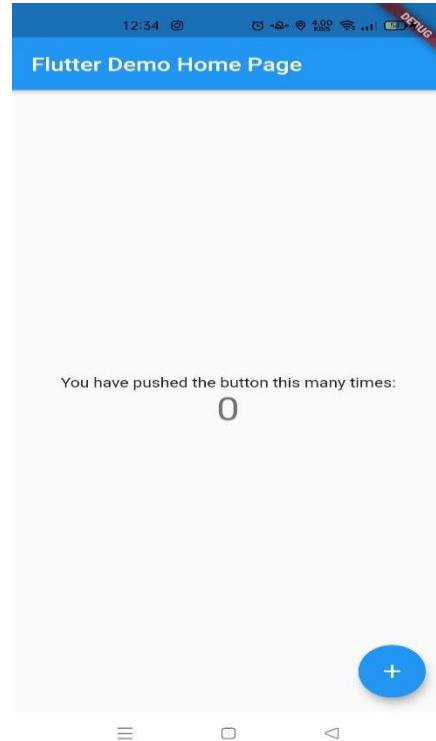
30. Hasil flutter doctor untuk kedua kalinya

31. Running Flutter App



The screenshot shows the Android Studio interface with the project 'flutter_app' open. The main.dart file is displayed in the code editor. The Run tab is selected, showing the command 'Running Gradle task "assembleDebug"...' and the output 'Flutter supports hot reload! Apply changes to your app in place, instantly. Learn more'. The bottom status bar indicates the device is connected as CPH2209 (mobile) and the app is running on it.

```
4 runApp(MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   // This widget is the root of your application.
9   @override
10  Widget build(BuildContext context) {
11    return MaterialApp(
12      title: 'Flutter Demo',
13      theme: ThemeData(
14        // This is the theme of your application.
15        //
16        // Try running your application with "flutter run". You'll see the
17        // application has a blue toolbar. Then, without quitting the app, try
18        // changing the primarySwatch below to Colors.green and then invoke
19        // "hot reload" (press "r" in the console where you ran "flutter run",
20        // or simply save your changes to "hot reload" in a Flutter IDE).
21        // Notice that the counter didn't reset back to zero; the application
22        // is not restarted.
23        primarySwatch: Colors.blue,
24      ),
25    );
26  }
27}
28
29
```



f. Hasil dan Pembahasan

1. Dokumentasi tugas berupa laporan secara individu tahap dan langkah-langkah dalam menginstallasi flutter.
2. Penjelasan pada laporan maksud dan langkah-langkah yang telah dilakukan.

g. Kesimpulan

Mahasiswa berhasil melakukan instalasi dan memahami maksud dari setiap langkah-langkah dalam instalasi flutter yang dilakukan pada prosedur kerja.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|--------------------|---|-------|---|---|---|
| 1 | Ketepatan waktu dan ketepatan dalam menjelaskan secara rinci command git yang digunakan pada revisi pertama | 1 | 2 | 3 | 4 |
| 2 | Ketepatan waktu dan ketepatan dalam menjelaskan command git yang digunakan pada revisi pertama | 1 | 2 | 3 | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat dalam menjelaskan tugas | 1 | 2 | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan dalam menjelaskan tugas | 1 | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 4 : PENGENALAN DAN WORKFLOW PADA GIT, FLUTTER & DART

| | |
|----------------------------------|---|
| Pokok Bahasan | : Pengelolaan Project pada Git |
| Acara Praktikum/Pertemuan | : Minggu 1 / 4 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu mengelola project pada git
2. Mahasiswa menjelaskan maksud membuat repository dan pengelolaan project pada git

b. Indikator

Keberhasilan mahasiswa dalam melakukan mengelola project pada git, serta memahami setiap command git revisi berikutnya.

c. Dasar Teori

-

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

Membuat project Git dan mengelolanya

Pastikan sudah install git di komputer anda. Jika belum terinstall silakan ikuti langkah-langkahnya di sini: <https://www.digitalocean.com/community/tutorials/how-to-contribute-to-open-source-getting-started-with-git>

Materi Video Hari Pertama

bagi teman teman yang membutuhkan materi video silahkan akses video diatas

Membuat Repository Baru

Repository atau repo di Gitlab biasa disebut dengan project. Berikut langkah-langkah untuk membuat repository baru:

- Setelah login anda akan menuju halaman Projects yang berisi semua repository yang anda miliki. Jika Anda user baru maka belum ada repository yang tersedia. Di halaman tersebut anda akan melihat tombol New Project di sebelah kanan atas. Klik tombol tersebut.

The screenshot shows the GitLab interface with the 'Projects' tab selected. At the top, there's a navigation bar with links for 'Projects', 'Groups', 'Activity', 'Milestones', and 'Snippets'. Below the navigation is a search bar and a 'New project' button. The main area is titled 'Your projects' with a count of 43. It includes filters for 'Starred projects', 'Explore projects', 'Filter by name...', and 'Last updated'. Underneath, there are two tabs: 'All' and 'Personal', with 'All' currently selected.

- Pada halaman membuat project baru, isi kolom project-name dengan nama yang Anda inginkan, misalkan: my first repo.
- Pada kolom visibility level Anda akan melihat tiga pilihan yaitu Private, Internal, dan Public. Private artinya repo yang akan kita buat bersifat rahasia, hanya Anda dan orang yang Anda berikan akses saja yang bisa membukanya. Sedangkan pilihan Internal berarti repo Anda akan dapat dilihat oleh semua user Gitlab. Pilihan “Public” berarti repo Anda akan terlihat oleh orang dari mana saja walaupun dia tidak memiliki akun gitlab. Pilih yang sesuai kebutuhan misalkan : Public.
- Di bagian bawahnya terdapat pilihan checkbox “Initialize repository with a README”. Centang saja jika anda ingin langsung clone repository nya setelah dibuat.

The screenshot shows the 'New project' creation form. On the left, there's a sidebar with information about what a project is and how to use CI/CD. The main form has four tabs at the top: 'Blank project' (selected), 'Create from template', 'Import project', and 'CI/CD for external repo'. The 'Project name' field contains 'my-first-repo'. The 'Project URL' field shows 'https://gitlab.com/mabdvh'. The 'Project slug' field also contains 'my-first-repo'. There's a note about creating a group for multiple projects. The 'Project description (optional)' field is empty. Under 'Visibility Level', the 'Private' option is selected. The 'Initialize repository with a README' checkbox is checked, with a note explaining it allows immediate cloning. At the bottom are 'Create project' and 'Cancel' buttons.

5. Jika berhasil maka Anda akan diarahkan menuju halaman project atau repo yang telah dibuat. Jika sebelumnya kita centang “Initialize repository with a README” maka repository kita akan berisi file README.md.
6. Pada Bagian kanan atas terdapat tombol clone. Ketika diklik maka akan keluar dua pilihan “Clone with SSH” dan “Clone with HTTPS”. Pilih saja clone dengan HTTPS. Klik tombol salin/copy supaya langsung otomatis tercopy pada clipboard. Jika ingin clone dengan SSH maka anda harus setting dulu akun anda agar terhubung dengan komputer Anda.

Muhamad Abduh > my-first-repo > Details

Project 'my-first-repo' was successfully created.

my-first-repo Project ID: 13167128

Add license 0 Commits 1 Branch 0 Tags 0 Bytes Files

Auto DevOps
It will automatically build, test, and deploy your application base
Learn more in the [Auto DevOps documentation](#)

Enable in settings

master my-first-repo + History Find file Web IDE

Initial commit Muhamad Abduh authored just now a66f21f3

7. Setelah itu menuju ke terminal atau cmd. Arahkan menuju folder dimana anda ingin menyimpan repository yang sudah Anda buat. Berikan perintah git clone <link_anda_yang_barusan_dicopy> (tanda <> hanyalah sebagai penanda template tidak usah ditulis di perintah cmd)

```
$ git clone <your_repository_link_url>
```

8. Jika sudah berhasil diclone maka akan ditemukan folder dengan nama repository di Gitlab. Cara mengeceknya dengan perintah “ls” atau “dir”.

```
$ ls
my-first-repo
```

Mengupload perubahan project dengan git push

Ketika kita sudah berhasil clone repository kita di Gitlab maka tentu kita ingin melakukan update atau penambahan code pada project kita. Agar perbaruan yang kita simpan di local (komputer) kita

dapat kita perbaharui juga di repository Gitlab maka kita bisa lakukan dengan git push. Berikut langkah-langkahnya:

1. Ketika kita sudah memberikan perubahan terhadap suatu file atau membuat folder-folder baru pada folder project kita maka cara mengecek status perbaruan adalah dengan git status

```
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
```

2. Setelah git status maka akan ditampilkan list file yang “modified” atau “untracked”. “modified” artinya adalah kita mengubah file yang sebelumnya ada sedangkan “untracked” artinya file tersebut sama sekali baru di repository. Dalam contoh di atas terdapat perubahan pada file “README.md”.

3. Untuk menaikkan perubahan (stage) yang kita buat maka berikan perintah git add

```
$ git add README.md

// Jika ada terdapat banyak file dan kita ingin naikkan semuanya maka perintahnya:
$ git add .
```

4. Sekarang perubahan tadi sudah pada tahap “staged”. jika kita cek status nya lagi maka akan didapati berbeda:

```
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

modified:   README.md
```

5. Selanjutnya adalah memberikan laporan atau commit terhadap perubahan yang kita buat dengan git commit.

```
$ git commit -m "Pesanan saya: Mengubah README.md"
```

6. tanda “-m” pada perintah commit artinya kita ingin memberikan pesan atau laporan terhadap perubahan yang sudah dibuat. Dalam hal ini laporannya adalah “Pesanan saya: mengubah README.md”

7. Selanjutnya adalah push untuk mengirim /upload perubahan yang ada di local computer kita menuju repository yang ada di Gitlab. Jika kita berada di branch bernama master maka perintahnya adalah:

```
$ git push origin master
```

Jika branch yang sedang kita miliki di local adalah branch lain selain master maka cara push nya adalah

```
$ git push origin <nama_branch>
```

Jika langkah tersebut sudah maka kita sudah berhasil mengupload perubahan yang kita buat di repository kita.

Git branch dan merge request (optional)

Membuat branch baru dan melakukan merge request

Clone repository

Terlebih dahulu lakukan clone terhadap repository project.

```
$ git clone https://gitlab.com/some-project  
$ cd some-project
```

Catatan: Link https di atas hanyalah contoh link dari sebuah repository. Link https tersebut bisa diperoleh di halaman gitlab pada bagian clone.

Buat branch baru

Setelah di-clone dan masuk ke folder project nya, buatlah branch baru . (myname hanyalah contoh nama branch, kita bisa mengubah sesuai nama yang diinginkan)

```
$ git branch myname  
$ git checkout myname
```

maka sekarang kita sudah berada di branch baru dan dapat mulai melakukan pekerjaan kita.

Catatan:

`myname` hanya contoh, silakan sesuaikan sendiri nama branch masing-masing.

Push ke branch

Setelah pekerjaan selesai, kita akan melakukan upload atau push ke branch kita sendiri.

```
$ git add .  
$ git commit -m "some message" // push ke branch  
$ git push origin myname
```

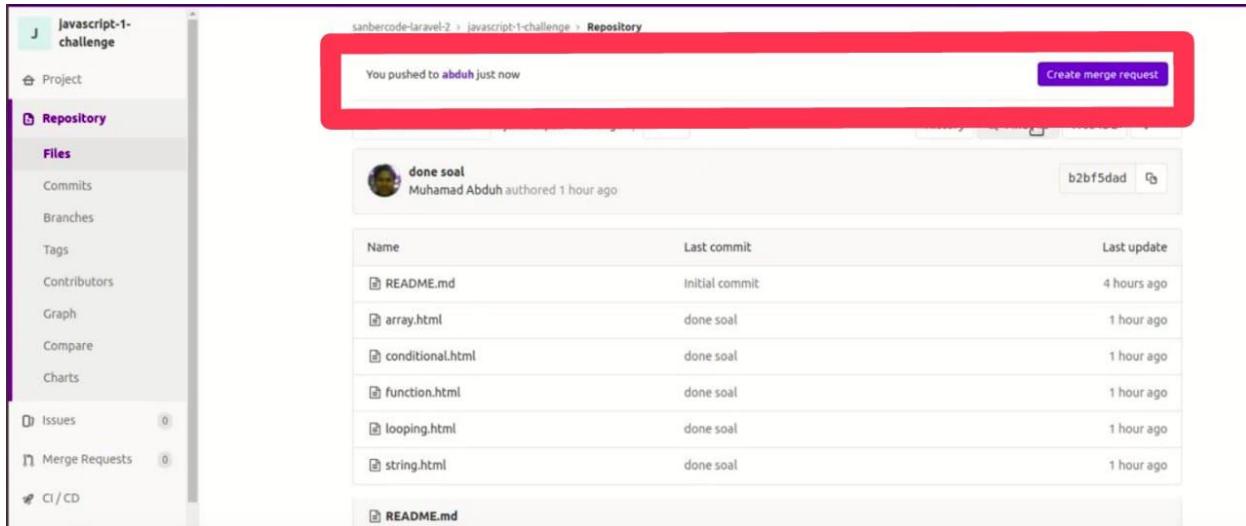
Catatan:

"`some message`" hanyalah contoh pesan atau komentar pada sebuah commit. Silahkan isi dengan jenis update yang dikerjakan.

Membuat merge request

Setelah melakukan push ke branch sendiri maka selanjutnya melakukan merge request.

Secara default pada halaman project akan tersedia tombol “create merge request”.



The screenshot shows a GitHub repository page for 'sanbercode-laravel-2 > javascript-1-challenge'. A red box highlights the 'Create merge request' button in the top right corner of the main content area. The left sidebar shows navigation options like Project, Repository, Files, Commits, Branches, Tags, Contributors, Graph, Compare, Charts, Issues (0), Merge Requests (0), and CI / CD. The main content area displays a commit message from 'done soal' (Muhamad Abdurrahman) made 1 hour ago, followed by a table of files with their last commit times.

| Name | Last commit | Last update |
|------------------|----------------|-------------|
| README.md | Initial commit | 4 hours ago |
| array.html | done soal | 1 hour ago |
| conditional.html | done soal | 1 hour ago |
| function.html | done soal | 1 hour ago |
| looping.html | done soal | 1 hour ago |
| string.html | done soal | 1 hour ago |

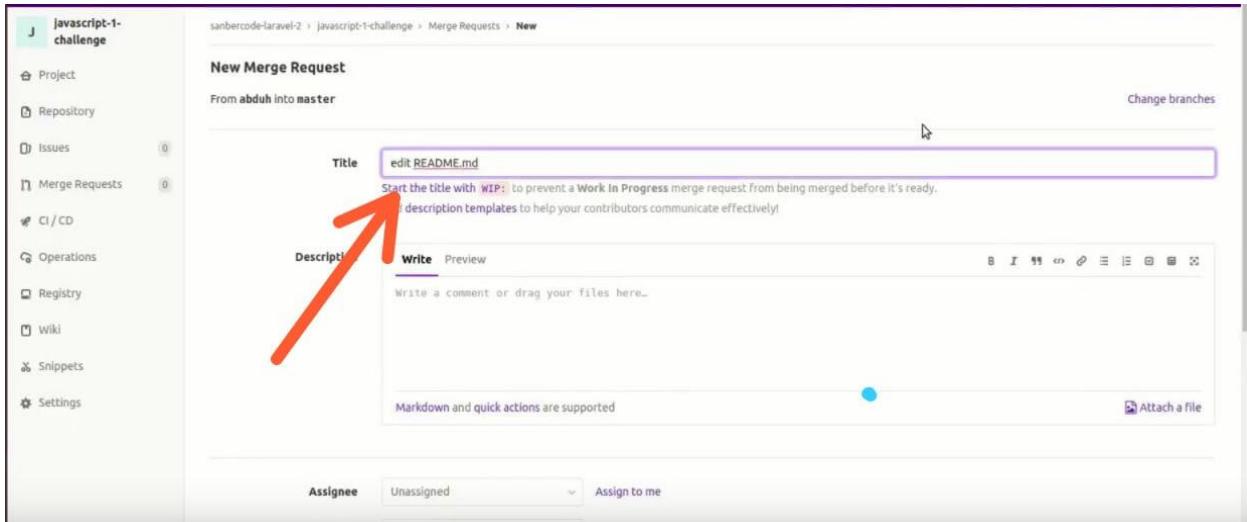
Jika tombol tersebut tersedia, klik tombol tersebut untuk menuju halaman merge request.

Pada halaman merge request isilah kolom title dengan pesan commit atau laporan progress pekerjaan misalkan : done, work in progress atau WIP, dsb.

Setelah itu, klik submit merge request maka pekerjaan kita sudah disubmit ke merge request dan akan direview oleh pengelola project.

Selengkapnya kamu bisa ikuti juga video tutorial berikut ini: tutorial merge request.

1. vscode , link download : <https://code.visualstudio.com/download>
2. Cmder, link download: <https://cmder.net/cmder>
3. Git bash,link download: <https://git-scm.com/downloads>, Git bash biasanya ter-install jika kita meng-install git di pc/laptop kita.



f. Hasil dan Pembahasan

1. Dokumentasi tugas berupa laporan secara individu tahap pengelolaan project pada git yang dibuktikan dengan screenshot pada masing-masing pekerjaannya.
2. Penjelasan pada laporan maksud dari setiap tahap dan langkah pengelolaan project pada git yang dilakukan.

g. Kesimpulan

Mahasiswa berhasil melakukan pengelolaan project pada git.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|--------------------|---|-------|-----|-----|-----|
| 1 | Ketepatan waktu dan ketepatan dalam menjelaskan secara rinci command git yang digunakan pada revisi kedua | 1 | 2 | 3 | (4) |
| 2 | Ketepatan waktu dan ketepatan dalam menjelaskan command git yang digunakan pada revisi kedua | 1 | 2 | (3) | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat dalam menjelaskan tugas | 1 | (2) | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan dalam menjelaskan tugas | (1) | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 5 : PEMROGRAMAN DART I

| | |
|----------------------------------|---|
| Pokok Bahasan | : Pemrograman Dart I (Intro & Tipe Data) |
| Acara Praktikum/Pertemuan | : Minggu 2 / 1 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami pemrograman dart.
2. Mahasiswa mampu memahami tipe data yang ada pada pemrograman dart.

b. Indikator

Keberhasilan mahasiswa pemrograman dart khususnya mengenai tipe data.

c. Dasar Teori

Dart adalah sebuah bahasa pemrograman yang dikembangkan oleh Google dan digunakan untuk membangun aplikasi mobile, desktop, backend dan web, bahasa pemrograman ini bertipe Object Oriented dimana struktur kode kita berada di dalam Class, Class berisi data dan method. Dart menggunakan C-style syntax yang mirip dengan C, Java, Javascript, Swift.

Menjalankan Dart

Ada Bermacam cara untuk menjalankan kode dart yang kita buat

1. Menjalankan dart di console browser
2. Menjalankan dart di tools online seperti <https://dartpad.dev/>
3. menjalankan dart dengan dart dev yang tersedia di website resmi dart

Menjalankan dart dengan dart dev

The screenshot shows a Dart code editor interface. On the left, there is a code editor window titled 'Dart' containing the following code:

```
1 void main() {  
2   print('Hello, World!');  
3 }
```

On the right, there is a 'Console' window showing the output of the script. Below the console, a message says 'no issues'.

Pada materi ini kita akan mencoba untuk menjalankan script dengan dart.

Dart Sdk adalah software berbasis bahasa pemrograman dart yang dijalankan di sisi server. Jika biasanya dart kita kenal erat kaitannya dengan client/browser tapi dengan Dart ini kita bisa membangun server menggunakan bahasa Dart.

Install Dart

Pertama-tama tentu kita harus menginstall terlebih dahulu dart sdk di komputer kita. Berikut ini link untuk download <https://gekorm.com/dart-windows/> (disarankan memilih versi STABLE). Untuk OS Windows dan macOs tinggal diikuti saja instalasinya sampai selesai, sedangkan untuk Ubuntu 18.04 kamu bisa install mengikuti <https://dart.dev/tools/sdk/archive>.

Untuk mengecek apakah instalasi dart sudah berhasil kita bisa jalankan script di terminal kita:

```
$ dart --version
```

Dart SDK version: 2.10.5 (stable) (Tue Jan 19 13:05:37 2021 +0100) on "windows_x64"

dengan perintah tersebut, diketahui saat ini terinstall dart dengan versi 2.10.5

```
Hello World
```

Untuk menjalankan dart mari kita coba buat file nya terlebih dahulu dengan menuliskan.

```
main.dart
```

```
void main(){  
  print("Hello World")  
}
```

lalu jika sudah masukan ke folder nya dan jalankan

```
$ dart main.dart
```

Demikian cara untuk menjalankan dart dengan dart sdk.

Dart I/O (input dan output)

di dalam pemerogrman dart kita juga dapat menggunakan input dan output, penggunaannya juga tergolong mudah dan cepat dengan menggunakan import 'dart.io'

contoh :

```
import 'dart:io';
void main(List<String> args) {
    print("masukan password: ");
    String inputText = stdin.readLineSync()!;
    print("password: " + inputText.toString());
}
```

jika code diatas error dapat di ganti untuk print nya menjadi seperti ini

```
import 'dart:io';
void main(List<String> args) {
    print("masukan password");
    String inputText = stdin.readLineSync()!;
    print("password: ${inputText}");
}
```

Data Type

Data Type atau dalam bahasa indonesia Tipe Data adalah sekumpulan informasi yang memiliki nilai dan karakteristik tertentu. Beberapa contoh tipe data pada dart di antaranya:

1. Number : tipe data angka
2. String : tipe data berupa text atau kumpulan karakter, biasanya string dibungkus dalam tanda petik ganda (double quote) atau tanda petik tunggal (single quote).
3. Boolean: tipe data dengan nilai true atau false
4. List & maps : daftar tipe data untuk merepresentasikan sekumpulan object

Variable

Variable adalah suatu blok data untuk menampung sekumpulan data dengan berbagai tipe data apapun. Dengan variable kita bisa menyimpan suatu nilai untuk kemudian kita olah kembali pada program kita. Untuk deklarasi variable dalam dart kita bisa gunakan sintaks var lalu diikuti nama variablenya.

```
void main() {
    var name = "John" ;// Tipe
    var angka = 12;
    var todayIsFriday = false ;
    print(name); // "John"
    print(angka) ;// 12
    print(todayIsFriday); // false
}
```

Waspada Pendeklarasian variabel yang tidak bernilai !

```
var items;  
print(items); //null
```

Operator

Operator adalah karakter khusus yang merepresentasikan sebuah tindakan. Operator terbagi ke dalam beberapa jenis:

1. Operator Aritmatika Operator yang melibatkan operasi matematika seperti tambah, kurang, kali, bagi.

- o Tambah (+)
- o Kurang (-)
- o Kali (*)
- o Bagi (/)
- o Modulus (%)

Modulus adalah sisa bagi. Contohnya $5 \% 3$ hasilnya adalah 2, $100 \% 5$ hasilnya 0.

2. Operator Assignment (`=`), Operator untuk mendaftarkan atau meng-assign suatu nilai ke dalam suatu variable

```
var angka;  
angka = 10; // Contoh assignment variable angka dengan nilai 10
```

3. Operator Perbandingan, Operator yang membandingkan suatu nilai dengan nilai yang lain. Hasil dari perbandingan ini akan dikembalikan dalam tipe data boolean `true` atau `false`.

- o Equal Operator (`==`)

```
var angka = 100;  
print(angka == 100); // true  
print(angka == 20); // false
```

- o Not Equal (`!=`)

```
var sifat = "rajin";  
print(sifat != "malas"); // true  
print(sifat != "bandel"); //true
```

- o Strict Equal (`===`) Selain membandingkan dua nilainya, strict equal juga membandingkan tipe datanya apakah sama atau tidak

```
var angka = 8;  
print(angka === "8"); // true, padahal "8" adalah string.  
print(angka === 8); // false, karena tipe data nya berbeda  
print(angka === 8); // true
```

- o Kurang dari & Lebih Dari (`<`, `>`, `<=`, `>=`)

```
var number = 17;  
print( number < 20 ); // true  
print( number > 17 ); // false  
print( number >= 17 ); // true, karena terdapat sama dengan  
print( number <= 20 ); // true
```

4. Operator Kondisional, Operator yang mengkombinasikan dua nilai kebenaran . Terdapat operator AND (`&&`) dan OR (`||`)

- o OR (||)

```

print(true || true); // true
print(true || false); // true
print(true || false || false); // true
print(false || false); // false

```
- o AND (&&)

```

print(true && true); // true
print(true && false); // false
print(false && false); // false
print(false && true && true); // false
print(true && true && true); // true

```

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

Mengenal lebih dalam tentang tipe data string pada dart

String

String adalah tipe data yang berisi karakter-karakter dibungkus dalam tanda petik ("") atau " "). Karakter-karakter pada suatu string dapat diakses dengan menggunakan indeks atau posisi karakter berada. Indeks pada string selalu mulai dari 0.

```

void main() {
  var sentences = "dart";
  print(sentences[0]); // "d"
  print(sentences[2]); // "r"
}

```

String pada dart juga memiliki property dan methods tertentu. Property dan methods tersebut bisa kita gunakan dalam memanipulasi data agar sesuai dengan program yang kita inginkan.

Numbers

int : adalah tipe data berdasarkan angka yang tidak membutuhkan koma di belakang

double : adalah tipe data yang spesifik sehingga membutuhkan koma

contoh :

```

void main() {
  // declare an integer
  int num1 = 10;

  // declare a double value
  double num2 = 10.50;

  // print the values
}

```

```
    print(num1); //10
    print(num2); //10.5
}
```

mengubah string menjadi integer, kita dapat mengubah tipe data di dart dengan menggunakan method num.parse()

contoh :

```
void main() {
    print(num.parse('12')); //12
    print(num.parse('10.91')); //10.91
}
```

dan jangan salah memasukan angka atau akan error

```
void main() {
    print(num.parse('12A'));
    print(num.parse('AAAA'));
}
```

kode ini akan menghasilkan error :

```
Unhandled exception:
FormatException: 12A
#0 num.parse (dart:core/num.dart:446)
#1 main (file:///D:/Demos/numbers.dart:4:13)
#2 _startIsolate.<anonymous closure> (dart:isolatepatch/isolate_patch.dart:261)
#3 _RawReceivePortImpl._handleMessage (dart:isolatepatch/isolate_patch.dart:148)
```

Mengubah dari int ke string

```
void main() {
    int j = 45;
    String t = "$j";
    print("hello"+ t);
}
```

Referensi Tambahan

https://www.tutorialspoint.com/dart_programming/dart_programming_data_types.htm

https://www.tutorialspoint.com/dart_programming/dart_programming_string.htm

https://www.tutorialspoint.com/dart_programming/dart_programming_boolean.htm

f. Hasil dan Pembahasan

Mahasiswa mampu memahami setiap tipe data yang ada pada pemrograman dart.

g. Kesimpulan

Mahasiswa berhasil memahami setiap tipe data yang ada pada pemrograman dart.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | | |
|--------------------|---|-------|---|---|---|--|
| 1 | Ketepatan waktu dan ketepatan menjelaskan detail dalam proses instalasi | 1 | 2 | 3 | 4 | |
| 2 | Ketepatan waktu dan ketepatan menjelaskan dalam proses instalasi | 1 | 2 | 3 | 4 | |
| 3 | Ketepatan waktu akan tetapi kurang tepat menjelaskan dalam proses instalasi | 1 | 2 | 3 | 4 | |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan menjelaskan dalam proses instalasi | 1 | 2 | 3 | 4 | |
| Jumlah skor | | | | | | |

ACARA 6 : PEMROGRAMAN DART I

| | |
|----------------------------------|---|
| Pokok Bahasan | : Pemrograman Dart I (Conditional) |
| Acara Praktikum/Pertemuan | : Minggu 2 / 2 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami pemrograman dart.
2. Mahasiswa mampu memahami struktur-struktur syntax conditional dalam pemrograman dart.

b. Indikator

Keberhasilan mahasiswa melakukan instalasi dan setting up gradle pada Android Studio.

c. Dasar Teori

Kondisional adalah sebuah metode yang melakukan pengecekan terhadap suatu pernyataan atau premis apakah bernilai benar atau tidak, jika benar maka akan menjalankan sebuah block code tertentu.

Operator Ternary

Operator ternary bisa dikatakan operator seperti `if-else` namun lebih pendek penulisannya, sebagai contoh disini kita akan mengecek apakah seseorang ini benar adalah `wahyu`, jika benar maka `print("wahyu")` jika tidak maka `print("bukan")`. Pertama kita gunakan `if-else` seperti dibawah ini.

```
void main() {
    var isThisWahyu = true;

    if(isThisWahyu) {
        print("wahyu");
    }else{
        print("bukan");
    }
}
```

Nah dengan ternary operator, kita dapat mempersingkatnya menjadi.

```
void main(){
    var isThisWahyu = true;
    isThisWahyu ? print("wahyu") : print("bukan");
}
```

Cukup mudah untuk dipahami, `?` digunakan untuk kondisi `true`, dan `:` digunakan sebagai pengganti `else`.

Kondisional dengan `if` / `else if` / `else`

1. Contoh 1 menjalankan kode jika premis bernilai `true`

```
void main() {  
    if ( true) {  
        print("jalankan code");  
    }  
}
```

2. Contoh 2 kode tidak dijalankan jika premis bernilai `false`

```
void main() {  
    if ( false ) {  
        print("Program tidak jalan code");  
    }  
}
```

3. Contoh 3 Premis dengan perbandingan suatu nilai

```
void main() {  
    var mood = "happy";  
    if ( mood == "happy" ) {  
        print("hari ini aku bahagia!");  
    }  
}
```

Di dalam kondisional dikenal juga dengan istilah branching atau percabangan. Dengan percabangan kita bisa mengecek nilai kebenaran dari berbagai premis yang kita sediakan. cara menambahkan kondisi atau premis lain dari premis pertamanya adalah dengan sintaks `else`. Contohnya ketika kita diminta untuk membeli telur dan buah ke minimarket oleh ibu kita maka akan banyak sekali kondisi atau kemungkinan yang terjadi dalam perjalanan kita membeli barang-barang tersebut di supermarket.

4. Contoh 4 Branching sederhana

```
void main() {  
    var minimarketStatus = "open";  
    if (minimarketStatus == "open") {  
        print("saya akan membeli telur dan buah");  
    } else {  
        print("minimarketnya tutup");  
    }  
}
```

Kondisi `else` di atas adalah kondisi selain `minimarketStatus == "open"`.

5. Contoh 5 Branching dengan kondisi

```
void main() {  
    var minimarketStatus = "close";  
    var minuteRemainingToOpen = 5;  
    if (minimarketStatus == "open") {  
        print("saya akan membeli telur dan buah");  
    } else if (minuteRemainingToOpen <= 5) {  
        print("minimarket buka sebentar lagi, saya tungguin");  
    } else {  
        print("minimarket tutup, saya pulang lagi");  
    }  
}
```

contoh di atas kita memberikan kondisi tambahan yaitu jika minimarket akan buka kurang atau sama dengan 5 menit lagi maka saya akan menunggu.

Selain Branching atau percabangan, kondisional juga dapat memiliki sarang atau tumpukan yaitu terdapat kondisional di dalam sebuah kondisional.

6. Contoh 6 Kondisional bersarang

```
void main() {
    var minimarketStatus = "open";
    var telur = "soldout";
    var buah = "soldout";
    if (minimarketStatus == "open") {
        print("saya akan membeli telur dan buah");
        if (telur == "soldout" || buah == "soldout") {
            print("belanjaan saya tidak lengkap");
        } else if (telur == "soldout") {
            print("telur habis");
        } else if (buah == "soldout") {
            print("buah habis");
        }
    } else {
        print("minimarket tutup, saya pulang lagi");
    }
}
```

pada kondisi di atas minimarketnya sudah buka lalu cek kondisi selanjutnya cek apakah telur atau buah habis. Karena kondisinya adalah jika telur ATAU buah salah satunya ada yang soldout maka belanjaan saya lengkap. Demikian seterusnya akan mengecek kondisi-kondisi yang disediakan dalam logika program yang dibuat.

Kondisional dengan **Switch Case**

Cara lain untuk melakukan pengecekan kondisi atau conditional adalah dengan **switch case**. Penggunaan switch case mirip seperti kita menyalakan tombol switch pada remote. Jika kondisi tombol yang dipijit adalah tombol dengan nomer tertentu maka akan menjalankan prorgram tertentu.

7. Contoh 7 Kondisional dengan **switch case**

```
void main() {
    var buttonPushed = 1;
    switch(buttonPushed) {
        case 1: { print('matikan TV!'); break; }
        case 2: { print('turunkan volume TV!'); break; }
        case 3: { print('tingkatkan volume TV!'); break; }
        case 4: { print('matikan suara TV!'); break; }
        default: { print('Tidak terjadi apa-apa'); }
    }
}
```

Pada kode di atas, switch akan mengevaluasi nilai **buttonPushed** jika **case** yang ditemui cocok dengan nilai yang diberikan maka kode pada **case** tersebut akan dijalankan sehingga pada console akan dimunculkan “**matikan TV!**“. Pada setiap case terdapat **break** yang berfungsi

untuk menghentikan proses switch agar tidak menjalankan case yang lain. Juga terdapat `default` yaitu kondisi dimana tidak ditemukan `case` yang sesuai.

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

Mahasiswa membuat laporan dokumentasi dengan penjelasan sedetail mungkin menggunakan syntax conditional pada pemrograman dart.

f. Hasil dan Pembahasan

Dokumentasi dalam bentuk laporan penggunaan syntax conditional pada pemrograman dart.

g. Kesimpulan

Mahasiswa berhasil membuat dokumentasi dalam bentuk laporan penggunaan syntax conditional pada pemrograman dart.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|--------------------|---|-------|-----|-----|-----|
| 1 | Ketepatan waktu dan ketepatan menjelaskan laporan dokumentasi detail dalam proses instalasi | 1 | 2 | 3 | (4) |
| 2 | Ketepatan waktu dan ketepatan menjelaskan laporan dokumentasi dalam proses instalasi | 1 | 2 | (3) | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat menjelaskan laporan dokumentasi dalam proses instalasi | 1 | (2) | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan menjelaskan laporan dokumentasi dalam proses instalasi | (1) | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 7 : PEMROGRAMAN DART I

| | |
|----------------------------------|---|
| Pokok Bahasan | : Pemrograman Dart I (Looping) |
| Acara Praktikum/Pertemuan | : Minggu 2 / 3 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami pemrograman dart.
2. Mahasiswa mampu mengenal beberapa struktur-struktur syntax looping dalam pemrograman dart.

b. Indikator

Keberhasilan mahasiswa mengenal dan menggunakan beberapa struktur-struktur syntax looping dalam pemrograman dart.

c. Dasar Teori

Loop/Iteration adalah tindakan mengulang / merepetisi sebuah proses, dengan tujuan untuk mendapatkan deret hasil, atau dengan tujuan mendapatkan hasil tertentu dengan repetisi. Setiap proses repetisi ini disebut sebagai Iteration atau Looping.

Untuk melakukan looping/iteration, Dart menyediakan beberapa jenis iteration, yaitu:

1. while-loop
2. for-loop

While-loop

While-loop adalah iterasi yang akan mengulang sebuah proses dengan sebuah kondisi tertentu.

Berikut adalah cara atau struktur penulisan while-loop:

```
while([Kondisi]) { // Kondisi yang menentukan apakah program akan melakukan iterasi.  
// Berupa boolean atau true/false.  
    [Proses] // Merupakan proses yang akan dijalankan dalam satu iterasi  
}
```

Pada while-loop, statement `while` akan mengambil sebuah nilai `true` atau `false` dari hasil kondisi yang diberikan. Apabila statement `while` mendapatkan nilai `true`, maka proses yang berada didalam curly bracket (`{ }`) akan dijalankan. Looping akan terus dilakukan selama kondisi while-loop masih `true`.

Untuk mencegah looping berjalan terus-menerus, dilakukan sebuah proses yang akan mengubah kondisi yang sedemikian rupa yang bertujuan untuk menghentikan looping dengan menghasilkan kondisi yang `false`.

For-loop

For-loop adalah bentuk lain dari iterasi, dimana statement `for` menjadi kontrol atas loop yang dilakukan. Hal ini yang menjadi pembeda antara for-loop dengan while-loop.

Berikut adalah cara atau struktur penulisan for-loop:

```
for([Inisialisasi], [Kondisi], [Incremental/Decremental]) {  
    [Proses] // Merupakan proses yang akan dijalankan dalam satu iterasi  
}
```

Pada for-loop, statement `for` akan menampung tiga parameter, yaitu sebut saja inisialisasi, kondisi, dan incremental/decremental. Ketiga parameter ini akan menjadi kontrol kapan loop ini harus berhenti. Pada parameter pertama, yaitu inisialisasi, sebuah variable diberikan nilai awal atau default. Pada parameter kedua, yaitu kondisi, for-loop akan terus berjalan selama kondisi ini masih terpenuhi, dengan kata lain, mengandung nilai `true`. Pada parameter ketiga, yaitu incremental/decremental, variabel yang menjadi kontrol terhadap loop ini akan diubah nilainya.

Best Practice: Walaupun memang for-loop dapat mengubah kondisi di dalam proses, namun best practice dari penggunaan for-loop adalah seluruh kendali atau kontrol dari looping ditentukan oleh variable yang diinisialisasi, di increment/decrement, dan juga kondisi for-loop pun menggunakan variable tersebut.

Untuk memudahkan kamu mendapatkan gambaran jelas tentang penggunaan for-loop, mari kita gunakan kedua contoh while-loop dan kita tulis ulang dalam bentuk for-loop.

Waspadai Infinite Looping!

Dengan sengaja atau tidak sengaja, kode kamu mungkin dapat menghasilkan infinite looping, atau looping yang tidak akan pernah berhenti. Bila ini terjadi, segera periksa statement kondisi kamu. contoh :

```
void main() {  
    var flag = 1;  
    while (flag < 10) {  
        // Loop akan terus berjalan, karena nilai flag tidak pernah berubah  
        print('Iterasi ke-' + flag.toString());  
    }  
}
```

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

Contoh Looping While-loop 1 Looping Angka 1-9 Sederhana

```
void main (){  
    var flag = 1;  
    while(flag < 10) {  
        print ("iterasi ke"+ flag.toString());  
        flag++; // Mengubah nilai flag dengan menambahkan 1  
    }  
}
```

Contoh Looping While-loop 2 Looping Mengembalikan Angka Total

```
void main() {  
    var deret = 5;  
    var jumlah = 0;  
    while (deret > 0) {  
        // Loop akan terus berjalan selama nilai deret masih di atas 0  
        jumlah += deret; // Menambahkan nilai variable jumlah dengan angka deret  
        deret--; // Mengubah nilai deret dengan mengurangi 1  
        print('Jumlah saat ini: ' + jumlah.toString());  
    }  
    print(jumlah);  
}
```

Contoh Looping For-loop 1 Looping Angka 1-9 Sederhana

```
void main(){  
for(var angka = 1; angka < 10; angka++) {  
    print('Iterasi ke-' + angka.toString());  
}  
}
```

Contoh Looping For-loop 2 Looping Mengembalikan Angka Total

```
void main() {  
    var jumlah = 0;  
    for (var deret = 5; deret > 0; deret--) {  
        jumlah += deret;  
        print('Jumlah saat ini: ' + jumlah.toString());  
    }  
    print('Jumlah terakhir: ' + jumlah.toString());  
}
```

Contoh Looping For-loop 3 Looping Dengan Increment dan Decrement Lebih dari 1

```
void main() {  
    for (var deret = 0; deret < 10; deret += 2) {  
        print('Iterasi dengan Increment counter 2: ' + deret.toString());  
    }  
}
```

```

print('-----');
for (var deret = 15; deret > 0; deret -= 3) {
    print('Iterasi dengan Decrement counter : ' + deret.toString());
}
}

```

f. Hasil dan Pembahasan

Dokumentasi dalam bentuk laporan penggunaan syntax looping pada pemrograman dart.

g. Kesimpulan

Mahasiswa berhasil membuat project yang menggunakan syntax looping pada pemrograman dart.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|--------------------|---|-------|-----|-----|-----|
| 1 | Ketepatan waktu dan ketepatan menjelaskan detail dalam proses pembuatan project | 1 | 2 | 3 | (4) |
| 2 | Ketepatan waktu dan ketepatan menjelaskan dalam proses pembuatan project | 1 | 2 | (3) | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat menjelaskan dalam proses pembuatan project | 1 | (2) | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan menjelaskan dalam proses pembuatan project | (1) | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 8 : PEMROGRAMAN DART I

| | |
|----------------------------------|---|
| Pokok Bahasan | : Pemrograman Dart I (Tugas) |
| Acara Praktikum/Pertemuan | : Minggu 2 / 4 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami pemrograman dart.
2. Mahasiswa mampu menyelesaikan tugas dalam pemrograman dart.

b. Indikator

Keberhasilan mahasiswa menyelesaikan tugas dalam pemrograman dart.

c. Dasar Teori

-

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

Mahasiswa membuat laporan dokumentasi dengan penjelasan dalam pembuatan project pertama pada Android Studio.

Pada tugas kali ini buatlah folder baru (Tugas-3 – Conditional) yang berisikan file dengan nama conditional.dart(untuk Tugas Conditional). Simpan folder tugas ini di dalam folder repository project yang dikerjakan pada BootcampFlutter.

Ternary operator

Petunjuk : untuk membuat sebuah kondisi ternary dimana anda akan diminta untuk menginstall aplikasi dengan jawaban Y/T , jadi tugas teman teman sekrang adalah memberi jawaban y/t saat ada input mau diinstall aplikasi, apabila (y)maka akan menampilkan "anda akan menginstall aplikasi dart", jika (T) maka akan keluar pesan "aborted" (Gunakan I/O).

If-else if dan else

Petunjuk : Kita akan memasuki dunia game werewolf. Pada saat akan bermain kamu diminta memasukkan nama dan peran . Untuk memulai game pemain harus memasukkan variable `nama` dan `peran`. Jika pemain tidak memasukkan nama maka game akan mengeluarkan warning “`Nama harus diisi!`“. Jika pemain memasukkan nama tapi tidak memasukkan peran maka game akan mengeluarkan warning “`Pilih Peranmu untuk memulai game`“. Terdapat tiga peran yaitu penyihir, guard, dan werewolf. Tugas kamu adalah membuat program untuk mengecek input dari pemain dan hasil response dari game sesuai input yang dikirimkan.

Petunjuk:

1. Nama dan peran diisi dengan i/o dan bisa diisi apa saja
2. Nama perlu dicek persis sesuai dengan input/output
3. Buat kondisi if-else untuk masing-masing peran

Input:

```
var nama = "John"  
var peran = ""
```

Output:

```
// Output untuk Input nama = " dan peran = " " , apabila kosong semua Nama harus diisi!"
```

```
//Output untuk Input nama = 'John' dan peran = " "apabila cuman diisi nama akan muncul Halo  
John, Pilih peranmu untuk memulai game!"
```

```
//Output untuk Input nama = 'Jane' dan peran 'Penyihir' "Selamat datang di Dunia Werewolf, Jane"  
"Halo Penyihir Jane, kamu dapat melihat siapa yang menjadi werewolf!"
```

```
//Output untuk Input nama = 'Jenita' dan peran 'Guard' "Selamat datang di Dunia Werewolf,  
Jenita" "Halo Guard Jenita, kamu akan membantu melindungi temanmu dari serangan werewolf."
```

```
//Output untuk Input nama = 'Junaedi' dan peran 'Werewolf' "Selamat datang di Dunia Werewolf,  
Junaedi" "Halo Werewolf Junaedi, Kamu akan memakan mangsa setiap malam!"
```

(Gunakan I/O)

Switch case

Kamu adalah seorang penyair yang setiap hari memberikan quotes yang sangat berguna untuk orang- orang sekitar mu buatlah sebuah switch case yang dapat menampung quotes tiap harinya :

Senin :

Segala sesuatu memiliki kesudahan, yang sudah berakhir biarlah berlalu dan yakinlah semua akan baik-baik saja.

Selasa :

Setiap detik sangatlah berharga karena waktu mengetahui banyak hal, termasuk rahasia hati.

Rabu :

Jika kamu tak menemukan buku yang kamu cari di rak, maka tulislah sendiri.

Kamis:

Jika hatimu banyak merasakan sakit, maka belajarlah dari rasa sakit itu untuk tidak memberikan rasa sakit pada orang lain.

Jumat:

Hidup tak selamanya tentang pacar.

Sabtu :

Rumah bukan hanya sebuah tempat, tetapi itu adalah perasaan.

Minggu :

Hanya seseorang yang takut yang bisa bertindak berani. Tanpa rasa takut itu tidak ada apapun yang bisa disebut berani.

(Gunakan I/O)

Switch Case

Kamu akan diberikan sebuah data dalam tiga variabel, yaitu `hari`, `bulan`, dan `tahun`. Disini kamu diminta untuk membuat format tanggal. Misal tanggal yang diberikan adalah hari 1, bulan 5, dan tahun 1945. Maka, output yang harus kamu proses adalah menjadi 1 Mei 1945.

Gunakan `switch case` untuk kasus ini, tidak perlu menggunakan i/o!

Contoh:

```
var hari = 21;
var bulan = 1;
var tahun = 1945;
// Maka hasil yang akan tampil di console adalah: '21 Januari 1945';
```

Skeleton Code / Code yang dicontohkan yang perlu diikuti dan dimodifikasi

```
var tanggal; // assign nilai variabel tanggal disini! (dengan angka antara 1 - 31)
var bulan;
// assign nilai variabel bulan disini! (dengan angka antara 1 - 12)
var tahun;
// assign nilai variabel tahun disini! (dengan angka antara 1900 - 2200)
```

Looping While

Pada tugas ini kamu diminta untuk melakukan looping dalam dart dengan menggunakan syntax `while`. Untuk membuat tantangan ini lebih menarik, kamu juga diminta untuk membuat suatu looping yang menghitung maju dan menghitung mundur. Jangan lupa tampilkan di console juga judul ‘LOOPING PERTAMA’ dan ‘LOOPING KEDUA’.”

Output:

```
LOOPING PERTAMA
2 - I love coding
4 - I love coding
6 - I love coding
8 - I love coding
10 - I love coding
12 - I love coding
14 - I love coding
16 - I love coding
18 - I love coding
20 - I love coding
LOOPING KEDUA
20 - I will become a mobile developer
18 - I will become a mobile
developer
16 - I will become a mobile developer
14 - I will become a mobile developer
12 - I will become a mobile developer
10 - I will become a mobile developer
8 - I will become a mobile developer
6 - I will become a mobile developer
4 - I will become a mobile developer
2 - I will become a mobile developer
```

Looping menggunakan for

Pada tugas ini kamu diminta untuk melakukan looping dalam dart dengan menggunakan syntax `for`. Untuk membuat tantangan ini lebih menarik, kamu juga diminta untuk memenuhi syarat tertentu yaitu:

SYARAT:

1. Jika angka ganjil maka tampilkan Santai
2. Jika angka genap maka tampilkan Berkualitas

3. Jika angka yang sedang ditampilkan adalah kelipatan 3 DAN angka ganjil maka tampilkan I Love Coding.

```
OUTPUT
1 - Santai
2 - Berkualitas
3 - I Love Coding
4 - Berkualitas
5 - Santai
6 - Berkualitas
7 - Santai
8 - Berkualitas
9 - I Love Coding
10 - Berkualitas
11 - Santai
12 - Berkualitas
13 - Santai
14 - Berkualitas
15 - I Love Coding
16 - Berkualitas
17 - Santai
18 - Berkualitas
19 - Santai
20 - Berkualitas
```

Membuat Persegi Panjang

Kamu diminta untuk menampilkan persegi dengan dimensi 8×4 dengan tanda pagar (#) dengan perulangan atau looping. Looping boleh menggunakan syntax apa pun (`while`, `for`, `do while`).

Output:

```
#####
#####
#####
#####
#####
```

Membuat Tangga

Kali ini kamu diminta untuk menampilkan sebuah segitiga dengan tanda pagar (#) dengan dimensi tinggi 7 dan alas 7. Looping boleh menggunakan syntax apa pun (`while`, `for`, `do while`).

Output:

```
#
##
###
####
#####
#####
#####
#####
```

f. Hasil dan Pembahasan

Dokumentasi dalam bentuk laporan hasil pengajaran tugas pada materi pemrograman dart.

g. Kesimpulan

Mahasiswa berhasil membuat dokumentasi dalam bentuk laporan penggerjaan tugas pada materi pemrograman dart.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|--------------------|--|-------|-----|-----|-----|
| 1 | Ketepatan waktu dan ketepatan menjelaskan laporan dokumentasi detail dalam pembuatan project | 1 | 2 | 3 | (4) |
| 2 | Ketepatan waktu dan ketepatan menjelaskan laporan dokumentasi dalam pembuatan project | 1 | 2 | (3) | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat menjelaskan laporan dokumentasi dalam pembuatan project | 1 | (2) | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan menjelaskan laporan dokumentasi dalam pembuatan project | (1) | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 9 : PEMROGRAMAN DART II

| | |
|----------------------------------|---|
| Pokok Bahasan | : Pemrograman Dart II (Function) |
| Acara Praktikum/Pertemuan | : Minggu 3 / 1 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami pemrograman dart.
2. Mahasiswa mampu mengenal struktur function dalam pemrograman dart.

b. Indikator

1. Keberhasilan mahasiswa dalam menerapkan linear layout pada studi kasus.
2. Keberhasilan mahasiswa dalam menerapkan relative layout pada studi kasus.

c. Dasar Teori

Function adalah sebuah blok kode yang disusun sedemikian rupa untuk menjalankan sebuah tindakan. Blok kode ini dibuat untuk dapat bisa digunakan kembali. Cara atau bentuk penulisan function adalah sebagai berikut:

```
function nama_function(parameter 1, parameter 2, ...) {  
    [Isi dari function berupa tindakan]  
    return [expression];  
}
```

Kode di atas tidak dapat kita copy-paste kan langsung, melainkan hanya sebuah bentuk penulisan `function`. Sebuah `function`, umumnya melakukan tindakan dan sebelum `function` berakhir, `function` bisa mengembalikan nilai dengan cara menambahkan sintaks `return`.

Kita juga dapat mengirimkan nilai ke dalam sebuah `function` dengan mencantumkannya ke dalam tanda kurung dalam penulisan `function`. Untuk mengirimkan nilai lebih dari satu, gunakan tanda , sebagai pemisah.

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

Contoh Function 1: Function sederhana tanpa return

```
void main() {
    tampilkan();
}

tampilkan() {
    print("Hello Peserta Bootcamp");
}
```

Contoh Function 2: Function sederhana dengan return

```
void main() {
    print(munculkanangka());
}

munculkanangka() {
    return 2;
}
```

Contoh Function 3: Function dengan parameter

```
void main() {
    print(kalikanDua(6));
}

kalikanDua(angka) {
    return angka * 2;
}
```

Contoh Function 4: Pengiriman parameter lebih dari satu

```
void main() {
    print(kalikan(5, 6));
}

kalikan(x, y) {
    return x * y;
}
```

Contoh Function 5: Inisialisasi parameter dengan nilai default

```
void main() {
    tampilkanangka(5);
}

tampilkanangka(n1, {s1: 45}) {
    print(n1); //hasil akan 5 karena initialisasi 5 didalam value tampilkan
    print(s1); //hasil adalah 45 karena dari parameter diisi 45
}
```

Waspadai pengiriman parameter yang UNDEFINED!

Kita juga dapat menampung function sebagai variable dengan sebuah bentuk function yang dinamakan Anonymous Function.

```
void main() {
    print(functionPerkalian(5, 6));
}

functionPerkalian(angka1, angka2) {
    return angka1 * angka2;
}
```

f. Hasil dan Pembahasan

Dokumentasi dalam bentuk laporan hasil implementasi function pada materi pemrograman dart.

g. Kesimpulan

Mahasiswa berhasil membuat dokumentasi dalam bentuk laporan penerapan function pada materi pemrograman dart.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|--------------------|---|-------|-----|-----|-----|
| 1 | Ketepatan waktu dan ketepatan dalam menjelaskan secara detail setiap baris code pada linear layout dan relative layout | 1 | 2 | 3 | (4) |
| 2 | Ketepatan waktu dan ketepatan dalam menjelaskan dengan singkat setiap baris code pada linear layout dan relative layout | 1 | 2 | (3) | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat dalam menjelaskan setiap baris code pada linear layout dan relative layout | 1 | (2) | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan dalam menjelaskan setiap baris code pada linear layout dan relative layout | (1) | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 10 : PEMROGRAMAN DART II

| | |
|----------------------------------|---|
| Pokok Bahasan | : Pemrograman Dart II (OOP) |
| Acara Praktikum/Pertemuan | : Minggu 3 / 2 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami pemrograman dart.
2. Mahasiswa mampu mengenal dan menerapkan konsep OOP dalam pemrograman dart.

b. Indikator

1. Keberhasilan mahasiswa dalam memahami pemrograman dart..
2. Keberhasilan mahasiswa dalam mengenal dan menerapkan konsep OOP dalam pemrograman dart.

c. Dasar Teori

Secara singkat class dapat diartikan blueprint dari suatu objek, maksudnya adalah kita mendesain suatu objek berdasarkan class yang kita buat, sebagai contoh dalam dunia nyata, jika kita ingin membuat sebuah objek meja maka kita perlu menggambar rancangan meja tersebut, dengan demikian kita akan mulai membuat meja berdasarkan rancangan tersebut, jika rancangannya baik maka hasilnya baik, begitu juga sebaliknya, jadi class dapat kita analogikan sebagai rancangannya dan hasil dari rancangan tersebut adalah objeknya.

1. Mendefenisikan Class

Untuk membuat kelas kita gunakan keyword class yang telah disediakan oleh bahasa Dart. Perhatikan contoh berikut:

```
class Dadu {  
}
```

Pertama kita mulai dengan keyword class, diikuti dengan nama kelasnya yaitu Dadu, kemudian tubuh kelas diawali dengan buka kurung kurawal "{" dan diakhiri dengan tutup kurung kurawal "}" . Jika sebuah kelas telah dibuat, maka kita langsung dapat menciptakan objek dari kelas tersebut, tentu jika kelas kita seperti contoh kelas Dadu, maka objek tersebut tidak dapat melakukan apapun,

karena kita belum menambahkan atribut ataupun metode kedalamnya. untuk menciptakan objek pada pemrograman dart kita gunakan keyword new. Perhatikan contoh program berikut ini:

```
Dadu dd = new Dadu();
```

Diawali dengan nama kelas yaitu Dadu kemudian nama objeknya yaitu dd (kita bisa menentukan nama objek yang lain tidak harus dd) kemudian new diakhiri dengan nama kelas.

```
nama_kelas nama_objek = new nama_kelas();
```

2. Enkapsulasi (Pembungkusan)

Enkapsulasi (encapsulation) adalah sebuah metode untuk mengatur struktur class dengan cara menyembunyikan alur kerja dari class tersebut.

Struktur class yang dimaksud adalah property dan method. Dengan enkapsulasi, kita bisa membuat pembatasan akses kepada property dan method, sehingga hanya property dan method tertentu saja yang bisa diakses dari luar class. Enkapsulasi juga dikenal dengan istilah ‘information hiding’. Dengan enkapsulasi, kita bisa memilih property dan method apa saja yang boleh diakses, dan mana yang tidak boleh diakses. Dengan menghalangi kode program lain untuk mengubah property tertentu, class menjadi lebih terintegrasi, dan menghindari kesalahan ketika seseorang ‘mencoba’ mengubahnya. Programmer yang merancang class bisa menyediakan property dan method khusus yang memang ditujukan untuk diakses dari luar. Pada pemrograman dart tidak ada keyword untuk membuat attribut atau metode menjadi private atau pun public, untuk membuat attribut ataupun fungsi menjadi private, cukup tambahkan underscore “_” sebelum nama attribut ataupun metode.

3. Inheritance

Inheritance adalah konsep OOP dimana sebuah class dapat menurunkan property dan method yang dimilikinya kepada class lain. Konsep inheritance dipakai untuk memanfaatkan fitur code reuse, yakni menghindari terjadinya duplikasi kode program.

Dalam bahasa Indonesia, inheritance ini disebut juga sebagai pewarisan atau penurunan.

Konsep inheritance membuat sebuah struktur atau hierarchy class dalam kode program. Class yang akan diturunkan bisa disebut sebagai class induk (parent class), super class, atau base class.

Sedangkan class yang menerima penurunan bisa disebut sebagai class anak (child class), sub class, derived class atau heir class.

Tidak semua property dan method class induk akan diturunkan. Property dan method dengan hak akses private tidak akan diturunkan kepada class anak. Hanya property dan method dengan hak akses protected dan public saja yang bisa diakses dari class anak.

Jadi yang dimaksud dengan inheritance dari class Ultramen dan Godzilla adalah extends, dimana class Ultramen dan Godzilla mewarisi sifat dari Character yang dimana Character memiliki object berupa level point sehingga saat dikumpulkan di main.dart secara default Godzilla dan Ultramen dari mengisi dari levelPoint yang ada pada parentnya yaitu Character.

4. Polymorphism

Polymorphism adalah konsep di mana suatu objek yang berbeda-beda dapat diakses melalui interface yang sama. Sebuah objek yang polymorphic dapat beradaptasi dengan metode apapun yang diimplementasikan pada objek tersebut, dan setiap class memiliki interpretasinya tersendiri terhadap interfacenya.

(SubBab) Tambahan Constructor

Constructor adalah method khusus yang akan dijalankan secara otomatis pada saat sebuah objek dibuat (instansiasi), yakni ketika perintah “new” dijalankan.

Constructor biasa digunakan untuk membuat proses awal dalam mempersiapkan objek, seperti memberi nilai awal kepada property, memanggil method internal dan beberapa proses lain yang digunakan untuk mempersiapkan objek.

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

Mengubah Prosedural ke Class

Buatlah sebuah folder baru dengan nama soal1 lalu di dalamnya buat file baru yaitu soal1.dart, lalu dari sistem prosedural ini ubahlah menjadi class dan object sehingga dapat dipanggil luasnya berupa class dan object

```

void main(List<String> args) {
    double setengah, alas, tinggi;
    setengah = 0.5;
    alas = 20.0;
    tinggi = 30.0;

    var luasSegitiga = setengah * alas * tinggi ;

    print(luasSegitiga);
}

```

Enkapsulasi(Pembungkusan)

Buatlah folder baru di dalam Tugas-7-oop dengan nama Tugas-Enkapsulasi, Dengan menggunakan teknik engkapsulasi buatlah sebuah class yang menerima parameter untuk menghitung luas lingkaran, rule nya gunakan validasi apabila input nilai -(min) maka harus di di kalikan -1 agar nilai tetap positif, selain itu boleh menggunakan method getter dan setter atau hanya menggunakan property getter dan setter, jadi boleh dipilih salah satu.

main.dart

```

import 'lingkaran.dart';
void main(List<String> args) {
    //tulis coding disini
}

```

lingkaran.dart

```

class Lingkaran{
    //tulis coding disini
}

double get luas => // isi coding disini ;

```

Inheritance

Buatlah folder baru di dalam Tugas-7-oop dengan nama Tugas-Inheritance lalu di dalam folder tersebut beri lagi file pada soal ini buatlah 5 class dan 1 main, isinya adalah

1. armor_titan.dart

2. attack_titan.dart
3. beast_titan.dart
4. human.dart
5. main.dart
6. titan.dart

pada

class 1 armor_titan memiliki object string yaitu terjang()=>"dush.. dush.." ,

class 2 attack_titan memiliki object string yaitu punch()=>"blam.. blam.."

class 3 beast_titan memiliki object string yaitu lempar()=>"wush wush.." ,

class 4 human memiliki object string yaitu killAlltitan()=>" Sasageyo ... Shinzo Sasageyo..."

selanjutnya di file titan.dart

buatlah class titan yang memiliki _powerPoint; yang akan men-get dan men-set nilai dari powerPoint masing masing turunannya

pada kelas 1 - 4 akan inheritance dari class titan, jadi otomatis akan memiliki power point setiap class nya

pada main.dart isilah value powerPoint yang ada pada 5 class dan juga tampilkan masing masing powertpoint nya jika powerPoint di bawah 5 maka akan di cetak 5 dan juga cetak masing masing object yang ada pada class masing-masing yang terdapat pada child class 1-4.

Polymorism

Buatlah folder baru di dalam Tugas-7-oop dengan nama Tugas-Polymorism lalu di dalam folder tersebut beri lagi file class dengan nama bangun_datar, segitiga, lingkaran, persegi, dan juga file main.dart untuk menjalankan classnya

pada bangun_datar akan mereturn luas dan keliling,

pada class Lingkaran akan mengextend dari bangun_datar

setelah itu akan dimasukan object nya yaitu rumus keliling dan luas ,

class persegi juga sama, adanya object luas sisi * sisi dan keliling = 4 * sisi ,

class segitiga ada luas = 0.5 * alas * tinggi, keliling = a +b + t, segitiga yang digunakan yaitu siku-siku

class lingkaran ada luas = 3.14 * jari-jari * jari-jari, keliling = 2 * pi * jari-jari

dan main.dart akan memanggil object bangun_datar.dart, selain itu juga akan memanggil class segitiga, persegi dan lingkaran dan akan mencetak luas dan keliling;

Constructor

Buatlah folder baru di dalam Tugas-7-oop dengan nama Tugas-constructor, di folder ini berisi main.dart dan juga employee.dart di dalam class employee buatlah object yang berisi id, name , departement , lalu panggil ke main.dart sebagai object yang akan bisa diisi value nya isi value bebas boleh nama pribadi teman-teman.

Jika sudah selesai dikerjakan, silahkan push hasil pekerjaan Anda ke repository Gitlab masing-masing (gunakan repository project yang sama bootcampFlutter yang digunakan folder Tugas-6-oop).

f. Hasil dan Pembahasan

Dokumentasi dalam bentuk laporan hasil implementasi OOP pada materi pemrograman dart.

g. Kesimpulan

Mahasiswa berhasil membuat dokumentasi dalam bentuk laporan penerapan OOP pada materi pemrograman dart.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|--------------------|--|-------|---|---|---|
| 1 | Ketepatan waktu dan ketepatan dalam menjelaskan secara detail setiap baris code pada constraint layout dan frame layout | 1 | 2 | 3 | 4 |
| 2 | Ketepatan waktu dan ketepatan dalam menjelaskan dengan singkat setiap baris code pada constraint layout dan frame layout | 1 | 2 | 3 | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat dalam menjelaskan setiap baris code pada constraint layout dan frame layout | 1 | 2 | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan dalam menjelaskan setiap baris code pada constraint layout dan frame layout | 1 | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 11 : PEMROGRAMAN DART II

| | |
|----------------------------------|---|
| Pokok Bahasan | : Pemrograman Dart II (Async & Sync) |
| Acara Praktikum/Pertemuan | : Minggu 3 / 3 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami pemrograman dart.
2. Mahasiswa mampu mengenal dan menerapkan konsep Asynchronous dan Synchronous dalam pemrograman dart.

b. Indikator

1. Keberhasilan mahasiswa memahami pemrograman dart.
2. Keberhasilan mahasiswa mengenal dan menerapkan konsep Asynchronous dan Synchronous dalam pemrograman dart.

c. Dasar Teori

Di dalam dunia pemrograman dart terdapat dua cara dalam menjalankan program: Synchronous dan Asynchronous. Synchronous artinya program berjalan secara berurutan sedangkan Asynchronous artinya program berjalan bersama-sama.

Terkadang di dalam program yang kita buat terdapat suatu sintaks yang mengharuskan code pada baris tersebut untuk dijalankan terlebih dahulu sebelum menjalankan sintaks pada baris selanjutnya. Hal ini dikenal dengan istilah blocking. Sebaliknya non-blocking artinya program berjalan dengan mengeksekusi sintaks dari baris ke baris secara paralel (bersama-sama) .

Perhatikan contoh program di bawah ini:

```
import 'dart:async';
void main (){
  print("saya dijalankan pertama");
  var timer = Timer(Duration(seconds: 3), ()=>print('saya dijalankan terakhir'));
  print("saya dijalankan kedua");
}
```

Jika kita jalankan program di atas, maka yang akan tampil terlebih dahulu di print adalah “saya dijalankan pertama” dan yang kedua adalah coding terakhir dan yang dicetak terakhir adalah var timer, walaupun ditulisa kedua karena proses async yang teradapat pada dart.

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

Synchronous to Asynchronous

Buatlah folder baru yaitu Tugas-9-AsyncAwaitFuture, setelah itu buat file baru untuk mengerjakan soal 1 yaitu soal1.dart kemudian ketikan coding di bawah ini, selanjutnya ubahlah coding synchronous berikut kedalam coding asynchronous dengan menggunakan await dan future, untuk output yang diharapkan yaitu saat di compile pertama munculnya yaitu luffy, zoro, killer dan nama character one piece selanjutnya 3 detik kemudian muncul get data [done] , name 3: hilmy

Nb: coding boleh di ubah- ubah sesuai kebutuhan dan logic dari teman-teman

```
void main(List<String> args) {  
    var h = Human();  
  
    print("Luffy");  
    print("zoro");  
    print("killer");  
    h.getData();  
    print(h.name);  
  
}  
  
class Human {  
    String name = "nama character one piece";  
  
    void getData(){  
        name = "hilmy";  
        print("get data [done]");  
    }  
}
```

f. Hasil dan Pembahasan

Dokumentasi dalam bentuk laporan hasil implementasi mengenal dan menerapkan konsep Asynchronous dan Syncronous pada materi pemrograman dart.

g. Kesimpulan

Mahasiswa berhasil membuat dokumentasi dalam bentuk laporan penerapan Asynchronous dan Syncronous pada materi pemrograman dart.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|--------------------|--|-------|-----|-----|-----|
| 1 | Ketepatan waktu dan ketepatan dalam menjelaskan secara detail setiap baris code pada table layout dan material design | 1 | 2 | 3 | (4) |
| 2 | Ketepatan waktu dan ketepatan dalam menjelaskan dengan singkat setiap baris code pada table layout dan material design | 1 | 2 | (3) | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat dalam menjelaskan baris code pada table layout dan material design | 1 | (2) | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan dalam menjelaskan baris code pada table layout dan material design | (1) | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 12 : PEMROGRAMAN DART II

Pokok Bahasan : Pemrograman Dart II (Async, Future, Await)

Acara Praktikum/Pertemuan : Minggu 3 / 4

Tempat : Daring / Luring Politeknik Negeri Jember

Alokasi Waktu : 100 menit

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami pemrograman dart.
2. Mahasiswa mampu mengenal dan menerapkan konsep Async, Future dan Await dalam pemrograman dart.

b. Indikator

1. Keberhasilan mahasiswa memahami pemrograman dart.
2. Keberhasilan mahasiswa mengenal dan menerapkan konsep Async, Future dan Await dalam pemrograman dart.

c. Dasar Teori

Future merupakan salah satu keyword yang disediakan oleh dart yang seperti namanya yaitu masa depan, jadi keyword ini gunanya untuk mengembalikan masa depan atau mengembalikan nilainya untuk waktu yang akan datang

contoh penggunaan future

```
void main() {  
    fetchUserOrder();  
    print('Ambil Datanya');  
}  
  
Future<void> fetchUserOrder() {  
    return Future.delayed(Duration(seconds: 2),  
        () => print('Selamat datang Peserta bootcamp flutter'));  
}
```

Bekerja dengan async dan await

Keyword Async dan Awair menyediakan cara deklarasi untuk mendefinisikan fungsi asynchronous dan menggunakan valuenya. dalam menggunakan async dan await ingat pedoman dasar ini :

1. sebelum menggunakan asynchronous terlebih dahulu tambahkan keywod async sebelum body
2. keyword await akan berfungsi apabila dia berada dalam fungsi async

3. dan berikut ini merupakan contoh yang mengkonversi dari fungsi synchronous ke dalam asynchronous

```
void main() async { ... }
```

jika fungsi memiliki tipe pengembalian yang di deklarasikan, maka perlu di perbaharui type menjadi Future<T>, dimana T adalah tipe nilai yang dikembalikan oleh fungsi, jika fungsi pengembalinya tidak secara eksplisit maka tipe data akan menjadi Future<void>

contoh :

```
Future<void> main() async { ... }
```

Sekarang Anda memiliki fungsi async, Anda dapat menggunakan kata kunci await untuk menunggu hingga Future selesai:

```
print(await createOrderMessage());  
  
Contoh Synchronous  
String createOrderMessage() {  
    var order = fetchUserOrder();  
    return 'Your order is: $order';  
}  
  
Future<String> fetchUserOrder() =>  
    // Imagine that this function is  
    // more complex and slow.  
    Future.delayed(  
        Duration(seconds: 2),  
        () => 'Large Latte',  
    );  
  
void main() {  
    print('Fetching user order...');  
    print(createOrderMessage());  
}
```

Contoh Asynchronous

```
Future<String> createOrderMessage() async {  
    var order = await fetchUserOrder();  
    return 'Your order is: $order';  
}  
  
Future<String> fetchUserOrder() =>  
    // Imagine that this function is  
    // more complex and slow.  
    Future.delayed(  
        Duration(seconds: 2),  
        () => 'Large Latte',  
    );  
  
Future<void> main() async {  
    print('Fetching user order...');  
    print(await createOrderMessage());
```

```
}
```

contoh asynchronous akan berbeda dalam tiga hal:

Jenis return/kembalian untuk createOrderMessage () berubah dari String ke Future <String>.

keyword async muncul sebelum body fungsi untuk createOrderMessage () dan main () .

keyword await muncul sebelum memanggil fungsi Asynchronous fetchUserOrder () dan createOrderMessage () .

Handling Error

untuk menangani handling error function async gunakan, try-catch jadi apabila succes akan masuk ke try dan jika error akan di tangani oleh catch

dan berikut ini merupakan contoh struktur coding try catch :

```
try {
  var order = await fetchUserOrder();
  print('Awaiting user order...');
} catch (err) {
  print('Caught error: $err');
}

Try-Catch contoh di dalam program
Future<void> printOrderMessage() async {
  try {
    var order = await fetchUserOrder();
    print('Awaiting user order...');
    print(order);
  } catch (err) {
    print('Caught error: $err');
  }
}

Future<String> fetchUserOrder() {
  // Imagine that this function is more complex.
  var str = Future.delayed(
    Duration(seconds: 4),
    () => throw 'Cannot locate user order');
  return str;
}

Future<void> main() async {
  await printOrderMessage();
}
```

Bekerja dengan synchronous dan asynchronous

dengan menggabungkan 2 blok ini kita dapat membuat ui kita menjadi lebih interaktif, shingga lebih memudahkan kita dalam menyajikan data yang interaktif, contoh penggunaan synchronous dan asynchronous sebagai berikut :

```

void main(List<String> args) async{
  var t = Titan(); // initialisasi t = object dari class titan

  print("zeke"); // mencetak zeke
  print(t.name); // akan mencetak string name yang pertama yaitu eren yeger
  await t.getName(); // masuk kedalam await untuk mendelayed object di dalamnya
  //dan akan mencetak get nem [done]
  print(t.name); //mencetak grisha yeger karena sudah di masukan kedalam async nya
  print("rener"); // akan mencetak rener

}

class Titan{
  String name = "Eren Yeger"; // mengisi value name eren yeger
  Future<void> getName()async{ // masuk kedalam Future dan object get name dan inisialisai async
    await Future.delayed(Duration(seconds: 3)); // memberi delayed pada object selama 3 detik
    name = "grisha"; //set name grisha
    print("get name [done]"); //cetak prin get name done
  }
}

```

Bekerja dengan future dan delayed tanpa async await

dengan menggunakan method future kita juga dapat membuat sebuah asynchrobnous dengan sama sama lebih mudah, berikut ini adalah contohnya

```

Future delayedPrint(int seconds, String message){
  final duration = Duration(seconds : seconds);
  return Future.delayed(duration).then((value) => message);
}

Run | Debug
main(List<String> args){
  print("roger");
  delayedPrint(2, "pirates"). then((status){
    print(status);
  });
  print("is");
}

```

Bekerja dengan async, await, future

dengan menggabungkan ketiganya kita dapat membuat aplikasi kita jauh lebih powerfull lagi dibanding kita harus bekerja dengan beberapa method sendiri-sendiri, untuk contoh penggunaanya sebagai berikut :

```

void main(List<String> args) async {
    print("Persiapan. mulai");
    print(await line());
    print(await line2());
}

Future<String> line() async {
    String greeting = "pagiku cerah matahari bersinar...";
    return await Future.delayed(Duration(seconds: 5), () => (greeting));
}

Future<String> line2() async {
    String greeting = "kugendong tas merahaku, di pundak ....";
    return await Future.delayed(Duration(seconds: 3), () => (greeting));
}

```

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

Future delayed

buatlah file baru dengan nama soal2.dart kemudian dengan menggunakan function future delayed buatlah output dari coding ini menjadi life is never flat, tidak perlu menggunakan async await

```

main(List<String> args){
    print("Life");
    print("never flat");
    print("is");
}

```

Async, Await and future

buatlah file baru dengan nama soal3.dart kemudian dengan menggunakan async await dan future ubahlah string berikut menjadi sebuah subtitle lagu, jadi awal masuk akan mencetak "Ready. sing" setelah itu ada jeda 5 detik kemudian akan muncul line() yang akan mencetak pernahkan kau merasa, setelah mencetak line() kemudian ada jeda waktu 3 detik untuk mencetak line2() yang berisi pernahkah kau merasa setelah mencetak line2() kemudian ada jeda waktu 2 detik untuk mencetak line3() yang berisi pernahkah kau merasa, dan terakhir yaitu memunculkan line4() dengan jeda 1 detik dari line 3 ke line 4 yang berisi, hatimu hampa pernahkah kau merasa hati mu kosong

```

void main(List<String> args) {
    print("Ready. Sing");
    line();
    line2();
    line3();
    line4();
}

line(){
    print("pernahkan kau merasa");
}
line2(){
    print("pernahkan kau merasa.....");
}

line3(){
    print("pernahkan kau merasa");
}
line4(){
    print("Hatimu hampa, pernahkan kau merasa hati mu kosong....");
}

```

f. Hasil dan Pembahasan

Dokumentasi dalam bentuk laporan hasil implementasi mengenal dan menerapkan konsep Async, Future dan Await pada materi pemrograman dart.

g. Kesimpulan

Mahasiswa berhasil membuat dokumentasi dalam bentuk laporan penerapan Async, Future dan Await pada materi pemrograman dart.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|--------------------|---|-------|---|---|---|
| 1 | Ketepatan waktu dan ketepatan dalam menyelesaikan studi kasus dalam bentuk project, serta menjelaskan setiap baris code. | 1 | 2 | 3 | 4 |
| 2 | Ketepatan waktu dan ketepatan dalam menyelesaikan studi kasus dalam bentuk project, namun kurang tepat menjelaskan setiap baris code. | 1 | 2 | 3 | 4 |
| 3 | Ketepatan waktu dan namun kurang tepat dalam menyelesaikan studi kasus dalam bentuk project serta menjelaskan setiap baris code. | 1 | 2 | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan dalam menjelaskan setiap baris code. | 1 | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 13 : MOCK UP & WIDGET CORE COMPONENT

| | |
|----------------------------------|---|
| Pokok Bahasan | : Mock Up (Tahap Sketsa/Konsep) |
| Acara Praktikum/Pertemuan | : Minggu 4 / 1 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami mockup dan pembuatannya.
2. Mahasiswa mampu menerapkan perancangan dalam bentuk mockup.

b. Indikator

Keberhasilan mahasiswa dalam memahami dan menerapkan mockup dalam perancangan.

c. Dasar Teori

Sebelum membuat aplikasi baik itu web based atau mobile biasanya akan dibuat terlebih dahulu sebuah prototipe atau desain mockup. Secara umum proses desain melingkupi : sketsa (level konsep) -> Wireframe (level komponen) -> Mockup/Prototype (level menghias/interaktif).

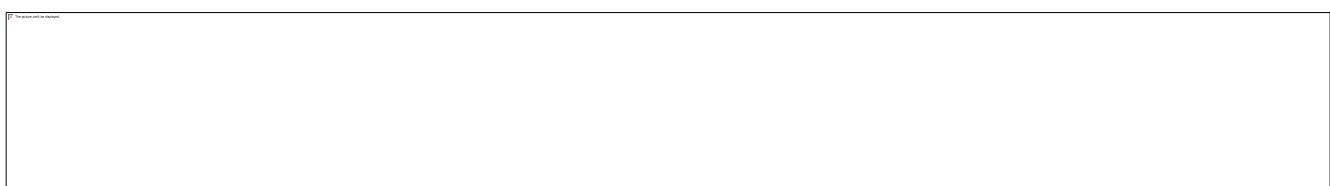
Tools untuk membuat Mockup yang direkomendasikan adalah Figma (Silahkan sign up / buat akun figma).

Tahap Sketsa (Konsep)

Tahap yang penting sebelum mendesain wireframe atau bahkan mockup adalah riset user. Pada bagian ini kita diminta untuk memetakan proses bisnis menjadi halaman-halaman (screens) yang dibutuhkan untuk mencapai tujuan (goals) tertentu.

Cara memetakan proses bisnis tersebut salah satunya dengan menggambar diagram alir (flow chart).

Contohnya sebuah aplikasi untuk pemesanan makanan seperti berikut:



Membuat sketsa

Tahap selanjutnya adalah mulai menggambar sketsa berdasarkan satu per satu step yang ada pada diagram alir menjadi sebuah sketsa kasar. Sketsa bisa dibuat dengan coretan di atas kertas atau bisa menggunakan tools sketsa sederhana di komputer.

Setiap sketsa harus berpedoman pada dua pertanyaan dasar:

1. Apa tujuan dari halaman/screen tersebut dibuat?
2. Bagaimana halaman/screen tersebut dapat membantu user mencapai tujuan(goal)?

Berikut ini contoh sebuah sketsa dari halaman Home aplikasi pemesanan makanan:



Jika diperhatikan sketsa di atas hanya digambarkan dengan blok-blok komponen berbentuk persegi sederhana saja.

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

Mockup aplikasi ini terdiri dari (minimal) 4 halaman. Kriteria dari masing-masing halaman adalah sebagai berikut:

Halaman Login/Register

Sesuai namanya, halaman ini merupakan halaman yang akan digunakan untuk melakukan kegiatan login/masuk ke aplikasi atau register/mendaftar ke aplikasi. Minimal 1 halaman (Halaman Login saja atau Halaman Register saja). Beberapa komponen yang harus ada pada halaman ini adalah:

1. Nama Aplikasi (bebas, boleh menggunakan Sanber App)
2. Logo/Icon Aplikasi (bebas, boleh hanya berupa bentuk kotak/lingkaran dengan background, atau bentuk lain yang dapat merepresentasikan logo)
3. Kotak input nama user / email
4. Kotak input password
5. Kotak input konfirmasi password (hanya untuk halaman register)
6. Tombol untuk login/register

Anda diperbolehkan menambahkan fitur atau komponen lain sesuai keinginan Anda.

Halaman About/Tentang Saya

Halaman ini berisikan informasi identitas atau profil Anda sebagai developer atau pembuat aplikasi (bukan untuk user). Data atau informasi yang digunakan pada halaman ini boleh bersifat fiktif (seperti pada data akun media sosial dan portofolio). Komponen yang harus ada dalam halaman ini adalah:

1. Foto
2. Nama Lengkap
3. Akun media sosial (minimal terdapat akun facebook, twitter, dan instagram)
4. skill yang di kuasai

Anda boleh menambahkan fitur atau komponen lain yang Anda inginkan.

Halaman Dashboard (List Jobs)

Halaman ini berisi daftar perusahaan yang tersedia/ yang telah membuka lowongan dengan spesifikasi sebagai berikut ini

1. halaman dashboard berisi tentang list yang isinya judul posisi yang diawarkan, nama perusahaan yang ingin merekrut, alamat perusahaan yang ingin merekrut dan range salary yang diberikan
2. boleh di tambah dengan search bar atau filter berdasarkan jobs

Halaman splash screen

1. halaman ini bisa berisi tentang splash screen atau tampilan awal untuk sebuah aplikasi saat muncul pertama kali

2. pemilihan warna dan logo silahkan bebas dari source mana saja

Anda boleh menambahkan fitur atau komponen lain yang Anda inginkan.

Jika sudah selesai mengerjakan, silahkan mengirimkan link share file figma Anda ke sanbercode.com. Link share tersebut dapat ditemukan pada bagian kanan atas, dengan menekan tombol "Share" dan klik pada bagian "copy link". Pastikan link dapat diakses oleh semua yang memiliki link (set "Anyone with link" dan "can view").

Apabila butuh inspirasi silahkan lihat halaman berikut ini :

Example mockup

1. Splash screen



2. HomeScreen

9:41

Let's Find Your Dream Jobs

Search a job or position

Jobs For You

See All

| | |
|--|-----------|
| Product Design Google Fulltime Junior California \$ 1.000 / month | Singapore |
|--|-----------|

Recently Posted

See All

| | |
|---|---|
| Digital Marketing 1-3 year Experience Fulltime Senior | Content Creator 1 year Experience Fulltime Internship |
| FrontEnd Dev 1-2 year Experience Fulltime Junior | UX Designer 1 year Experience Fulltime Junior |

Freelance Jobs

| | |
|---------------------------------|------------------------------------|
| Glints HR Recruiter Fulltime | LinkedIn Software E... Fulltime |
|---------------------------------|------------------------------------|

Home Bookmarks Application My Account

f. Hasil dan Pembahasan

Dokumentasi dalam bentuk laporan hasil mockup dan pembuatannya, serta perancangan dalam bentuk mockup.

g. Kesimpulan

Mahasiswa memahami mockup dan pembuatannya, serta perancangan dalam bentuk mockup.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|--------------------|--|-------|-----|-----|-----|
| 1 | Ketepatan waktu dan ketepatan dalam menjelaskan secara detail perbedaan listview, spinner, dan auto complete text view | 1 | 2 | 3 | (4) |
| 2 | Ketepatan waktu dan ketepatan dalam menjelaskan dari tugas | 1 | 2 | (3) | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat dalam menjelaskan tugas | 1 | (2) | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan dalam menjelaskan tugas | (1) | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 14 : MOCK UP & WIDGET CORE COMPONENT

| | |
|----------------------------------|---|
| Pokok Bahasan | : Mock Up (Tahap Wireframe) |
| Acara Praktikum/Pertemuan | : Minggu 4 / 2 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami wireframe dan pembuatannya.
2. Mahasiswa mampu menerapkan perancangan dalam bentuk wireframe.

b. Indikator

Keberhasilan mahasiswa dalam menerapkan perancangan dalam bentuk wireframe.

c. Dasar Teori

Tahap Wireframe

Tahap selanjutnya setelah membuat sketsa adalah menggambar wireframe berdasarkan sketsa yang sudah dibuat pada tahap konsep. Membuat wireframe artinya menggambarkan komponen-komponen pada sketsa dengan komponen yang lebih detail misalkan : komponen input (form), komponen button, komponen text, atau komponen gambar dan lain-lain.

Memilih Frame

Agar lebih aktual dengan ukuran sebenarnya, gunakanlah frame sesuai dengan yang diinginkan. Misalkan pilihlah device Android atau iPhone.

Figma

Pada tahap membuat Mockup diperlukan aplikasi yang dapat membantu komunikasi antara Client, Designer, dan Developer. Salah satu tools yang bisa digunakan adalah Figma (tools lain yang umum digunakan adalah Sketch, Adobe XD, Adobe Ilustrator, dan lain-lain).

Terdapat beberapa kelebihan dari Figma dibandingkan dengan aplikasi/tools lainnya, antara lain:

1. Salah satunya adalah kemampuannya dalam melakukan kolaborasi antar desainer. Figma memungkinkan 2 atau lebih UI Designer untuk berkerja sama secara langsung dalam satu waktu.

2. Dan karena figma berbasis web, sehingga user dari figma tidak harus melakukan proses instalasi aplikasinya.
3. Terlebih, figma cenderung memiliki banyak fitur yang dapat digunakan secara gratis.

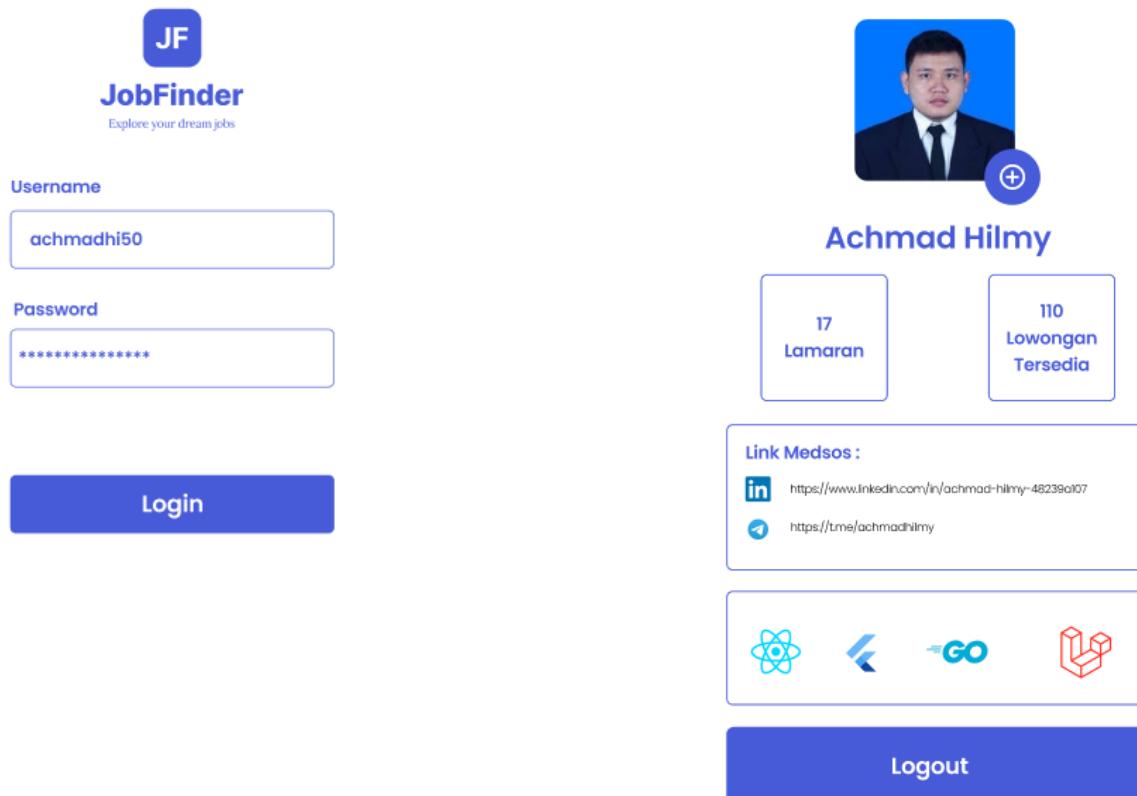
Untuk mengetahui bagaimana cara menggunakan Figma terutama dalam membangun suatu desain aplikasi mobile, dapat langsung mengunjungi bagian tugas-9-mockup

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

1. Login / Register
2. about me



f. Hasil dan Pembahasan

Dokumentasi dalam bentuk laporan hasil wireframe dan pembuatannya, serta perancangan dalam bentuk wireframe.

g. Kesimpulan

Mahasiswa berhasil mengimplementasikan wireframe dan pembuatannya, serta perancangan dalam bentuk wireframe.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|--------------------|---|-------|-----|-----|-----|
| 1 | Ketepatan waktu dan ketepatan dalam menjelaskan secara detail setiap baris code pada percobaan listview | 1 | 2 | 3 | (4) |
| 2 | Ketepatan waktu dan ketepatan dalam menjelaskan baris code pada percobaan listview | 1 | 2 | (3) | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat baris code pada percobaan listview | 1 | (2) | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan baris code pada percobaan listview | (1) | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 15 : MOCK UP & WIDGET CORE COMPONENT

| | |
|----------------------------------|---|
| Pokok Bahasan | : Widget Core Component I |
| Acara Praktikum/Pertemuan | : Minggu 4 / 3 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu mengimplementasikan widget core component pada flutter.
2. Mahasiswa mampu memahami konsep mengimplementasikan widget core component.

b. Indikator

Keberhasilan mahasiswa dalam mengimplementasikan widget core component.

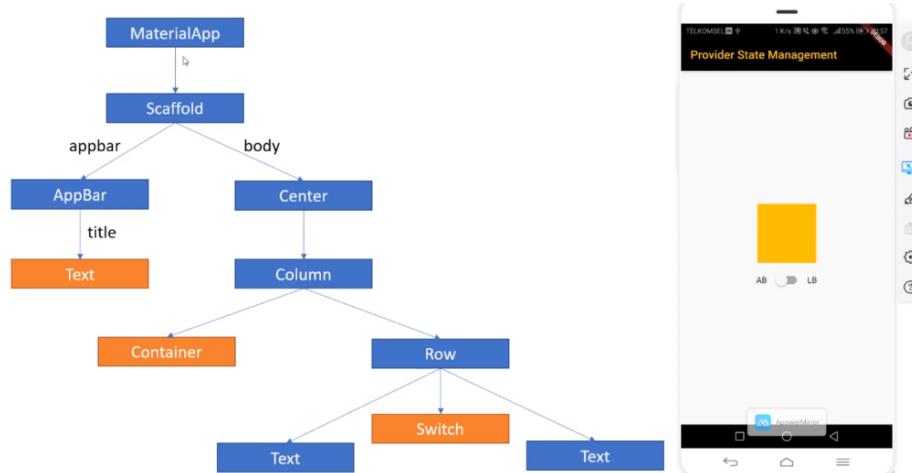
c. Dasar Teori

Widget Core Component

Bagi sobat sanber yang belum mengerti apa itu widget, sederhananya pada Flutter seluruh tampilan seperti tombol, gambar, teks, list, ikon, bahkan satu layar pada handphone tersebut merupakan sekumpulan dari banyak widget.

Struktur flutter apps

berikut ini merupakan struktur dari flutter apps/ per page halaman yang akan kita layout, umumnya akan ada scaffold, appbar, body dan didalam body ada widget-widget



pada warna biru adalah core widget/component yang terdapat pada flutter, semacam tag html, head, body, footer dst.

Properti pada Widget

Setiap widget memiliki property, misalnya kita membuat sebuah tombol, lalu kita ingin agar warna backgroundnya itu hijau, dan warna tulisannya putih, lalu teksnya kita tebalkan.

Untuk melakukan itu semua, maka tombol kita atur melalui propertinya.



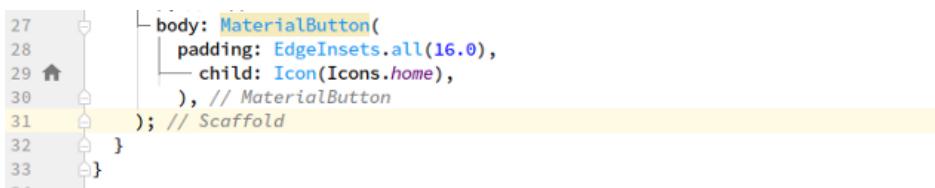
```
19 class MyHomePage extends StatelessWidget {
20   @override
21   Widget build(BuildContext context) {
22     return Scaffold(
23       appBar: AppBar(
24         title: Text("Aplikasi Pertamaku"),
25         backgroundColor: Colors.purple,
26       ), // AppBar
27     ); // Scaffold
28   }
29 }
```

Dari gambar tersebut kita bisa pelajari bahwa ciri dari widget dia diawali dengan huruf kapital dia berupa class, digambar itu juga kita memiliki widget Scaffold, AppBar, dan Teks.

Lalu widget Scaffold dia memiliki property appBar, sedangkan pada widget AppBar dia memiliki property title, dan backgroundColor.

Widget di dalam widget

Dalam membuat aplikasi nantinya tentu kita akan menggunakan banyak widget, pada Flutter setiap Widget umumnya memiliki properti **child** atau anak, dari properti inilah kita bisa menggunakan widget didalamnya.



```
27   body: MaterialButton(
28     padding: EdgeInsets.all(16.0),
29     child: Icon(Icons.home),
30   ), // MaterialButton
31 ); // Scaffold
32 }
33 }
```

Selain child, kita juga mengenal property **children** yang artinya anak-anak, sesuai dengan namanya, berarti widget tersebut bisa memiliki banyak widget-widget, sebagai contoh kita ingin menampilkan daftar menu makanan, maka tentu akan banyak widget yang berfungsi untuk menampilkan keterangan nama makanan didalamnya.

```
  body: Column(
    children: <Widget>[
        Text("Nasi + Mie Rebus"),
        Text("Nasi + Mie Rebus + Telor"),
        Text("Nasi + Ayam"),
        Text("Nasi + Ayam + Es Jeruk"),
        Text("Nasi + Ikan + Teh Manis"),
        Text("Nasi + Sate Ayam"),
        ],
    ),
)
```



General Widget

Ada banyak sekali widget-widget yang telah disediakan pada flutter, pada kali ini kita akan membahas widget apa saja yang paling penting untuk kita pahami saat pertama kali membuat aplikasi menggunakan Flutter ini.

Scaffold

Scaffold adalah widget utama untuk membuat sebuah halaman pada flutter, scaffold ini memiliki beberapa parameter yang biasa kita gunakan seperti appBar untuk membuat AppBar, body untuk bagian tubuhnya, atau kita juga bisa menambahkan floating action bar, maupun mengganti warna background bodynya.

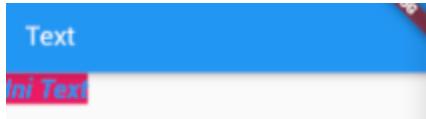
```
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text('Sample Code'),
    ),
    body: Center(
      child: Text('You have pressed the button ${_count} times.'),
    ),
    backgroundColor: Colors.blueGrey.shade200,
    floatingActionButton: FloatingActionButton(
      onPressed: () => setState(() => _count++),
      tooltip: 'Increment Counter',
      child: const Icon(Icons.add),
    ),
  );
}
```



Text

Text berfungsi untuk menampilkan sebuah teks biasa, atau bisa kita berikan style dengan menambahkan property style.

```
...
body: Text('Ini Text', style: TextStyle(
    color: Colors.blue,
    backgroundColor: Colors.pink,
    fontSize: 20.0,
    fontStyle: FontStyle.italic,
    fontWeight: FontWeight.bold
),)
...
```



Icon

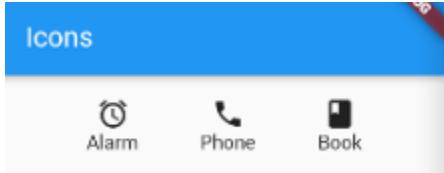
Widget ini untuk menggunakan icon yang telah disediakan, berikut adalah contohnya.

```
...
body: Container(
    padding: EdgeInsets.all(16.0),
    child: Row(
        mainAxisAlignment: MainAxisAlignment.spaceEvenly,
        children: <Widget>[
            Column(
                children: <Widget>[
                    Icon(Icons.access_alarm),
                    Text('Alarm')
                ],
            ),
        ],
    ),
)
```

```

),
Column(
  children: <Widget>[
    Icon(Icons.phone),
    Text('Phone')
  ],
),
Column(
  children: <Widget>[
    Icon(Icons.book),
    Text('Book')
  ],
),
),
],
),
)
...

```



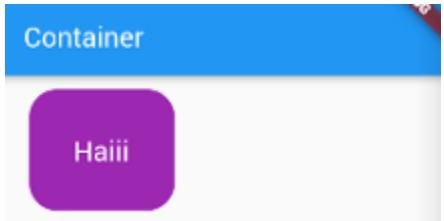
Container

Container merupakan widget yang fungsinya untuk membungkus widget lain sehingga dapat diberikan nilai seperti margin, padding, warna background, atau dekorasi.

```

...
body: Container(
  padding: EdgeInsets.all(32.0),
  margin: EdgeInsets.fromLTRB(20.0, 10.0, 20.0, 0),
  decoration: BoxDecoration(
    borderRadius: BorderRadius.circular(20.0),
    color: Colors.purple),
  // color: Colors.purple,
  child: Text('Haiii', style: TextStyle(color: Colors.white, fontSize: 20.0)),
)
...

```



Button

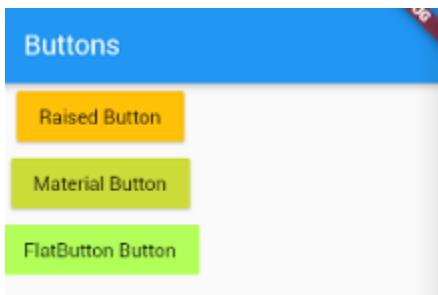
Terdapat 3 widget Button yang umumnya dipakai yaitu RaisedButton, MaterialButton, dan FlatButton

Pada raised button dan Material tombol akan sedikit menonjol.

Pada flat button tombolnya akan datar tanpa adanya efek-efek seperti bayangan dan lain-lain.

```
...
body: Column(
    children: <Widget>[
        RaisedButton(
            color: Colors.amber,
            child: Text("Raised Button"),
            onPressed: () {},
        ),
        MaterialButton(
            color: Colors.lime,
            child: Text("Material Button"),
            onPressed: () {},
        ),
        FlatButton(
            color: Colors.lightGreenAccent,
            child: Text("FlatButton Button"),
            onPressed: () {},
        ),
    ],
)
...

```



TextField

TextField merupakan widget yang berguna untuk membuat form untuk diisi user.

```
...
body: Padding(
    padding: const EdgeInsets.all(8.0),
    child: Form(
        child: Column(
            children: <Widget>[
                TextFormField(
                    decoration: InputDecoration(hintText: "Username"),
                ),
                TextFormField(
                    obscureText: true,
                    decoration: InputDecoration(hintText: "Password"),
                ),
                RaisedButton(
                    child: Text("Login"),
                    onPressed: () {},
                )
            ],
        ),
    ),
)
,
```



d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

-

f. Hasil dan Pembahasan

Dokumentasi dalam bentuk laporan hasil mengimplementasikan widget core component, dan juga memahami konsep mengimplementasikan widget core component.

g. Kesimpulan

Mahasiswa berhasil mengimplementasikan widget core component, dan juga memahami konsep mengimplementasikan widget core component.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | | |
|--------------------|---|-------|-----|-----|---|-----|
| 1 | Ketepatan waktu dan ketepatan implementasi pada percobaan card view dan recycler view dengan penjelasan singkat | 1 | 2 | 3 | 4 | (4) |
| 2 | Ketepatan waktu dan ketepatan implementasi pada percobaan card view dan recycler view | 1 | 2 | (3) | 4 | |
| 3 | Ketepatan waktu akan tetapi kurang tepat implementasi pada percobaan card view dan recycler view | 1 | (2) | 3 | 4 | |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan implementasi pada percobaan card view dan recycler view | (1) | 2 | 3 | 4 | |
| Jumlah skor | | | | | | |

ACARA 16 : MOCK UP & WIDGET CORE COMPONENT

| | |
|----------------------------------|---|
| Pokok Bahasan | : Widget Core Component II |
| Acara Praktikum/Pertemuan | : Minggu 4 / 4 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu mengimplementasikan widget core component pada flutter.
2. Mahasiswa mampu memahami konsep widget core component.

b. Indikator

Keberhasilan mahasiswa dalam mengimplementasikan widget core component.

c. Dasar Teori

Image

Image merupakan widget yang dapat digunakan untuk menampilkan gambar, bisa menggunakan internet atau dengan local foler

Internet

Image.network('https://cdn.pixabay.com/photo/2019/11/10/17/36/indonesia-4616370_1280.jpg');

Assets lokal

Gambar dari asset lokal yaitu file gambar yang ditambahkan kedalam folder aplikasi yang nantinya digunakan untuk menampilkan gambar tanpa koneksi internet. Tahapan menampilkan gambar lokal yaitu :

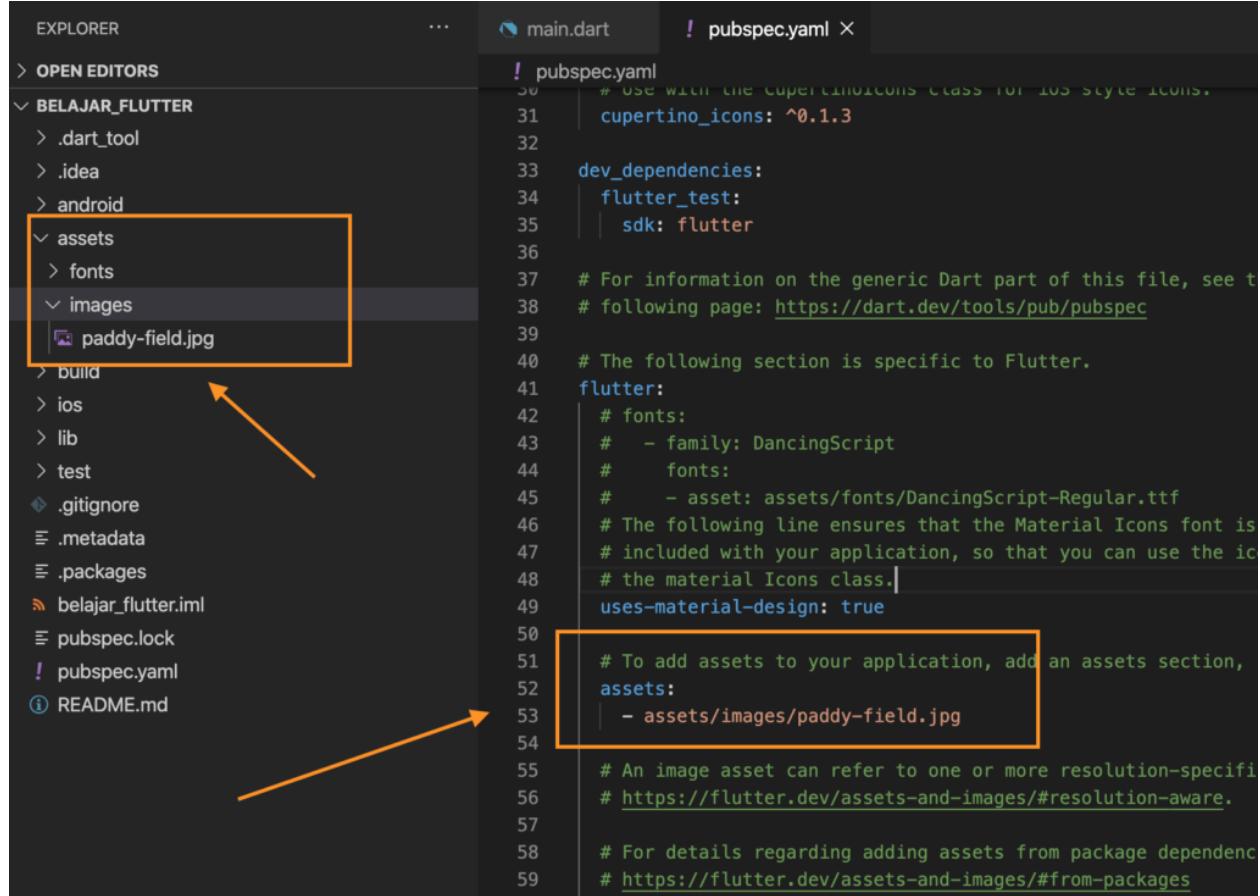
1. buat folder baru assets/images
posisi folder sejajar dengan **pubspec.yaml**. Untuk penamaan folder bisa apa aja namun sebaiknya gunakan nama yang umum pada istilah pemrograman. Hal tersebut berguna untuk membantu orang lain apabila bekerja sebagai tim.
2. Masukan gambar ke dalam folder tersebut
3. Registrasikan image / gambar yang tadi kita masukan dengan cara edit file **pubspec.yaml**

```
flutter:  
  assets:  
    - assets/images/paddy-field.jpg
```

Jika memiliki beberapa gambar yang ingin di import maka kita dapat menggunakan cara yang lebih sederhana yaitu hanya menggunakan nama direktori saja.

```
flutter:
```

```
  assets:  
    - assets/images/
```



```
EXPLORER          main.dart      ! pubspec.yaml ×  
> OPEN EDITORS  
└ BELAJAR_FLUTTER  
  > .dart_tool  
  > .idea  
  > android  
  > assets  
    > fonts  
    > images  
      paddy-field.jpg  
  > build  
  > ios  
  > lib  
  > test  
  ⚙ .gitignore  
  ⚙ .metadata  
  ⚙ .packages  
  📄 belajar_flutter.iml  
  ⚙ pubspec.lock  
  🎛 pubspec.yaml  
  ⓘ README.md  
  
! pubspec.yaml  
30  # USE WITH THE CupertinoIcons CLASS FOR IOS-STYLE ICONS.  
31  cupertino_icons: ^0.1.3  
32  
33  dev_dependencies:  
34    flutter_test:  
35      sdk: flutter  
36  
37  # For information on the generic Dart part of this file, see t  
38  # following page: https://dart.dev/tools/pub/pubspec  
39  
40  # The following section is specific to Flutter.  
41  flutter:  
42    # fonts:  
43    #   - family: DancingScript  
44    #     fonts:  
45    #       - asset: assets/fonts/DancingScript-Regular.ttf  
46    # The following line ensures that the Material Icons font is  
47    # included with your application, so that you can use the ic  
48    # the material Icons class.  
49    uses-material-design: true  
50  
51  # To add assets to your application, add an assets section,  
52  assets:  
53    - assets/images/paddy-field.jpg  
54  
55  # An image asset can refer to one or more resolution-specific  
56  # https://flutter.dev/assets-and-images/#resolution-aware.  
57  
58  # For details regarding adding assets from package dependenc  
59  # https://flutter.dev/assets-and-images/#from-packages
```

Harap perhatikan tab dan spasi karena apabila posisi value asset sejajar atau menjorok terlalu dalam akan error

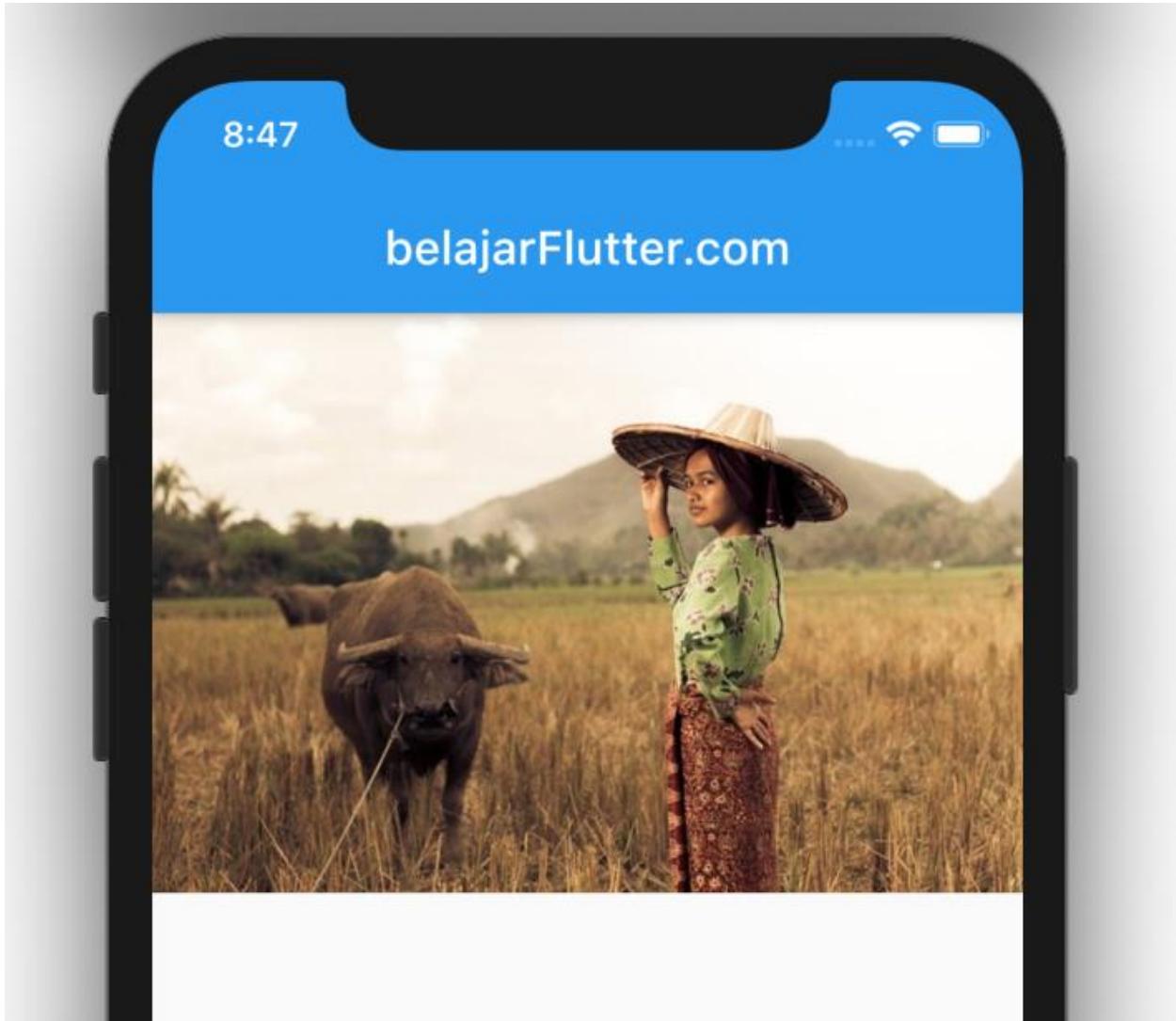
Tampilkan asset image dengan Image.asset Widget

```
Image.asset('assets/images/paddy-field.jpg');
```

Contoh lengkapnya seperti kode dibawah ini

```
import 'package:flutter/material.dart';  
  
void main() => runApp(BelajarImage());  
  
class BelajarImage extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      home: Scaffold(  
        appBar: AppBar(  
          title: Text("belajarFlutter.com"),  
        ),  
        body: Image.asset('assets/images/paddy-field.jpg'),  
      ),  
    );  
  }  
}
```

```
)  
);  
}  
}
```



Format gambar yang support yaitu : JPEG, WebP, GIF, animated WebP/GIF, PNG, BMP, and WBMP. Penggunaan local image biasanya untuk gambar yang bersifat statis seperti logo aplikasi, background asset. Semakin banyak gambar yang di import maka semakin besar pula ukuran aplikasinya nanti

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

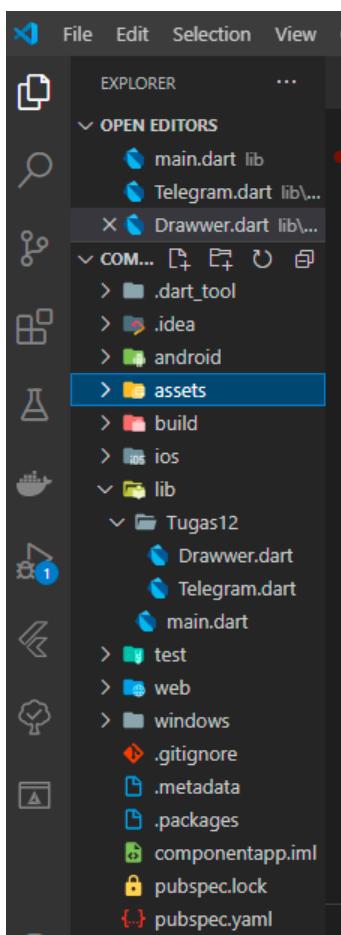
Mempersiapkan folder

Dalam mengerjakan tugas ini, silahkan membuat folder "Tugas" terlebih dahulu di dalam folder proyek Flutter yang telah dibuat. Dan di dalam folder "Tugas" buatlah folder "Tugas11". Dan di dalam folder "Tugas11", Anda bisa membuat file Telegram.dart dan DrawerScreen.dart dan Models untuk dummy datanya. Untuk lebih jelasnya ikuti video diatas.

Mendownload data yang diperlukan

Pada tutorial ini ada beberapa file yaitu models untuk dummy datanya dan juga foto profile, untuk foto profile gunakan foto masing-masing di taruh di folder assets/img, lalu untuk file models dapat di download di link berikut ini :

https://drive.google.com/drive/folders/1tf8h_AtzO6WqwNhi4yY0xNOPhNDaEVXd?usp=sharing



Tutorial Membuat UI Aplikasi Telegram

Berikut tutorial membangun UI aplikasi mobile Telegram. Ikuti langkah-langkah yang digunakan di video tersebut dan pastikan code yang diikuti dapat dijalankan pada perangkat Android Anda. Dan Bagi Teman-teman yang tidak bisa mengikuti video tutorial, dapat diikuti coding berikut :

main.dart

```
import 'package:flutter/material.dart';
import 'Tugas/Tugas12/Telegram.dart';
void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: Telegram(),
    );
  }
}
```

Telegram.dart

```
lib > Tugas > Tugas12 > Telegram.dart > _TelegramState > build
1   import 'package:flutter/material.dart';
2   import 'package:sanberappflutter/Tugas/Tugas12/DrawerScreen.dart';
3   import 'package:sanberappflutter/Tugas/Tugas12/Models/Chart_model.dart';
4
5   class Telegram extends StatefulWidget {
6     @override
7     _TelegramState createState() => _TelegramState();
8   }
9
10  class _TelegramState extends State<Telegram> {
11    @override
12    Widget build(BuildContext context) {
13      return Scaffold(
14        appBar: AppBar(title: Text("Telegram"),
15        actions: <Widget>[
16          Padding(padding: const EdgeInsets.all(8.0),
17            child: Icon(Icons.search),
18          ) // Padding
19        ], // <Widget>[]
20      ), // AppBar
21      drawer: DrawerScreen(),
22      body: ListView.separated(
23        itemBuilder: (ctx, i){
24          return ListTile(
25            leading: CircleAvatar(
26              radius: 28,
27              backgroundImage: NetworkImage(items[i].profileUrl),
28            ), // CircleAvatar
29            title: Text(items[i].name, style: TextStyle(fontWeight: FontWeight.bold),),
30            subtitle: Text(items[i].message),
31            trailing: Text(items[i].time),
32            ); // ListTile
33        },
34        separatorBuilder: (ctx, i){
35          return Divider();
36        }, itemCount: items.length), // ListView.separated
37        floatingActionButton: FloatingActionButton(
38          child: Icon(Icons.create, color: Colors.white),
39          backgroundColor: Color(0xFF65a9e0),
40          onPressed: (){},
41        ), // FloatingActionButton
42      ); // Scaffold
43    }
44 }
```

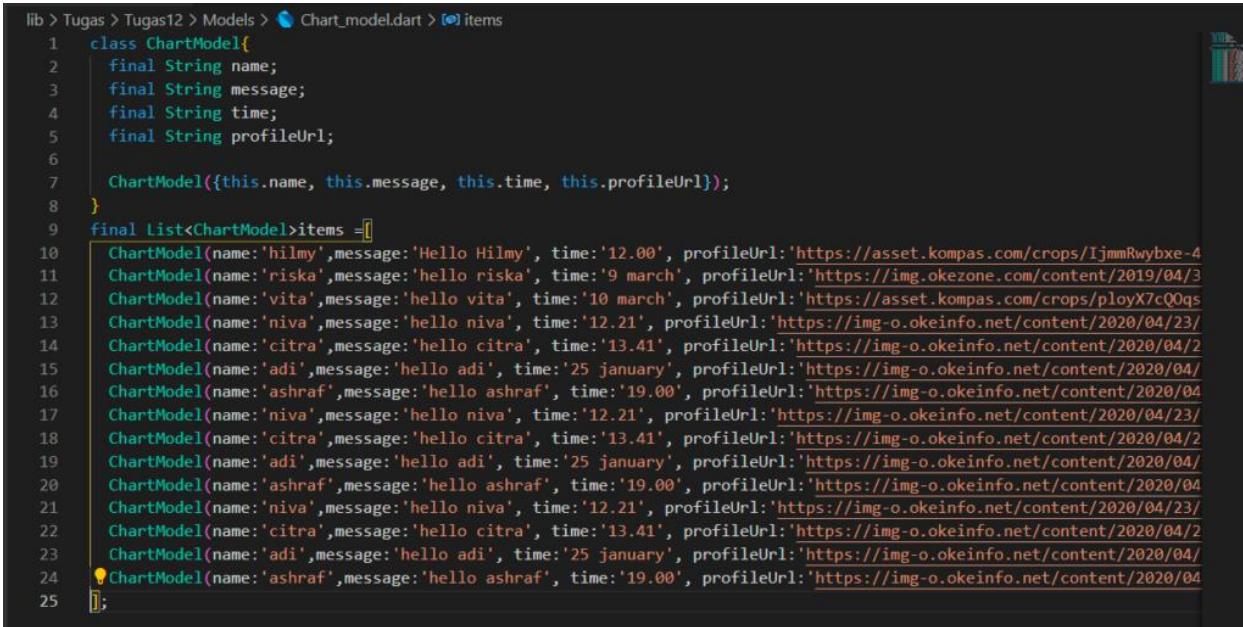
DrawerScreen.dart

```
1 import 'package:flutter/cupertino.dart';
2 import 'package:flutter/material.dart';
3
4 class DrawerScreen extends StatefulWidget {
5   @override
6   _DrawerScreenState createState() => _DrawerScreenState();
7 }
8
9 class _DrawerScreenState extends State<DrawerScreen> {
10   @override
11   Widget build(BuildContext context) {
12     return Drawer(
13       child: ListView(
14         children: <Widget>[
15           UserAccountsDrawerHeader(
16             accountName: Text("Achmad Hilmy Firdaus"),
17             currentAccountPicture: CircleAvatar(
18               backgroundImage: AssetImage("assets/img/hilmy.jpg")
19             ), // CircleAvatar
20             accountEmail: Text("achmad.hilmy@sanbercode.com"),
21           ), // UserAccountsDrawerHeader
22           DrawerListTile(
23             iconData : Icons.group,
24             title: "NewGroup",
25             onTilePressed: (){},
26           ), // DrawerListTile
```

```
27     DrawerListTile(  
28         IconData : Icons.lock,  
29         title: "New Secret Group",  
30         onTilePressed: (){},  
31     ), // DrawerListTile  
32     DrawerListTile(  
33         IconData : Icons.notifications,  
34         title: "New Channel Chat",  
35         onTilePressed: (){},  
36     ), // DrawerListTile  
37     DrawerListTile(  
38         IconData : Icons.contacts,  
39         title: "contacts",  
40         onTilePressed: (){},  
41     ), // DrawerListTile  
42     DrawerListTile(  
43         IconData : Icons.bookmark_border,  
44         title: "Saved Message",  
45         onTilePressed: (){},  
46     ), // DrawerListTile  
47     DrawerListTile(  
48         IconData : Icons.phone,  
49         title: "Calls",  
50         onTilePressed: (){},  
51     ), // DrawerListTile
```

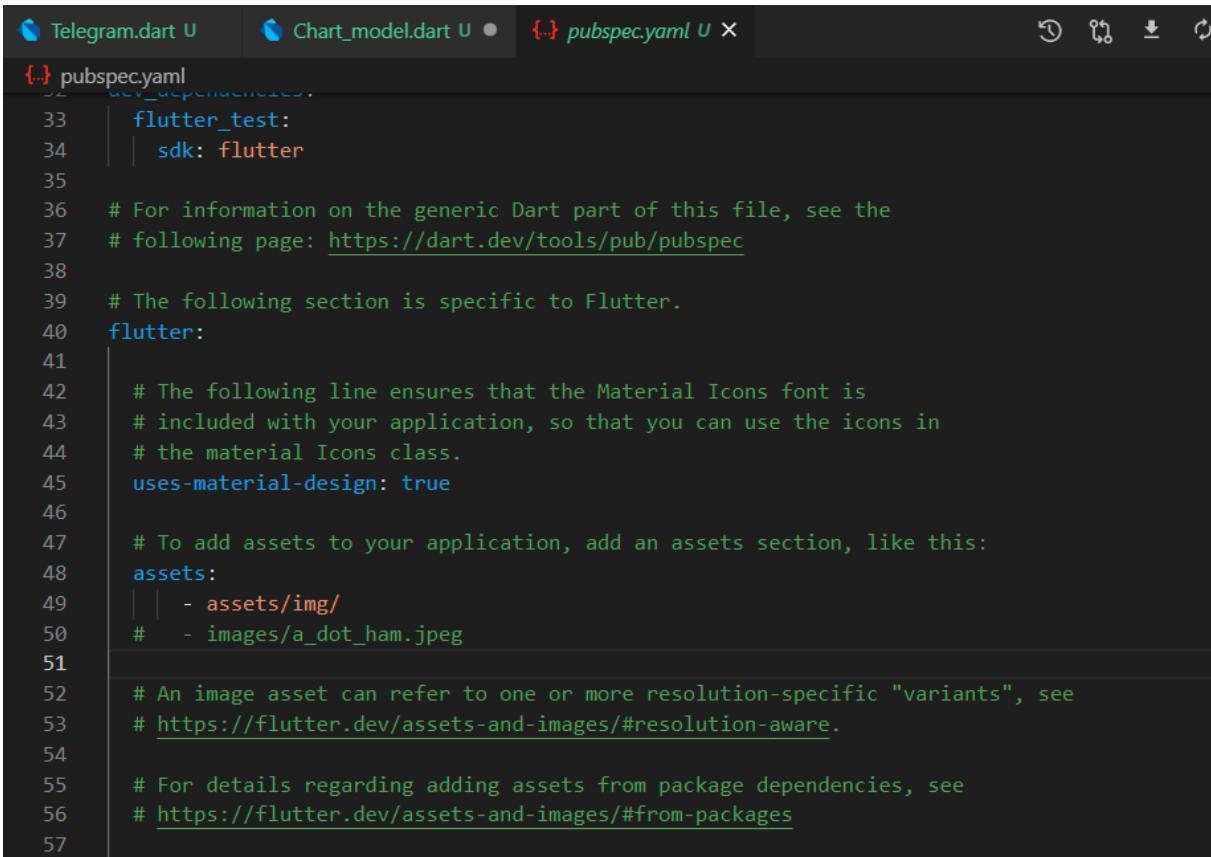
```
52     |   ], // <Widget>[]
53   | } // ListView
54   | ); // Drawer
55 }
56
57 class DrawerListTile extends StatelessWidget {
58   final IconData iconData;
59   final String title;
60   final VoidCallback onTilePressed;
61
62   const DrawerListTile({Key key, this.iconData, this.title, this.onTilePressed})
63   :super(key:key);
64   @override
65   Widget build(BuildContext context) {
66     return ListTile(
67       onTap: onTilePressed,
68       dense: true,
69       leading: Icon(iconData),
70       title: Text(title, style: TextStyle(fontSize: 16),),
71     ); // ListTile
72   }
73 }
74 }
```

Model/Chart_Models.dart (Sudah ada tinggal download)



```
lib > Tugas > Tugas12 > Models > Chart_model.dart > items
1  class ChartModel{
2    final String name;
3    final String message;
4    final String time;
5    final String profileUrl;
6
7    ChartModel({this.name, this.message, this.time, this.profileUrl});
8
9  final List<ChartModel>items =[  
10   ChartModel(name:'hilmy',message:'Hello Hilmy', time:'12.00', profileUrl:'https://asset.kompas.com/crops/IjmmRwybxe-4  
11   ChartModel(name:'riska',message:'hello riska', time:'9 march', profileUrl:'https://img.okezone.com/content/2019/04/3  
12   ChartModel(name:'vita',message:'hello vita', time:'10 march', profileUrl:'https://asset.kompas.com/crops/ployX7cQ0qs  
13   ChartModel(name:'niva',message:'hello niva', time:'12.21', profileUrl:'https://img-o.okeinfo.net/content/2020/04/23/  
14   ChartModel(name:'citra',message:'hello citra', time:'13.41', profileUrl:'https://img-o.okeinfo.net/content/2020/04/2  
15   ChartModel(name:'adi',message:'hello adi', time:'25 january', profileUrl:'https://img-o.okeinfo.net/content/2020/04/  
16   ChartModel(name:'ashraf',message:'hello ashraf', time:'19.00', profileUrl:'https://img-o.okeinfo.net/content/2020/04/  
17   ChartModel(name:'niva',message:'hello niva', time:'12.21', profileUrl:'https://img-o.okeinfo.net/content/2020/04/23/  
18   ChartModel(name:'citra',message:'hello citra', time:'13.41', profileUrl:'https://img-o.okeinfo.net/content/2020/04/2  
19   ChartModel(name:'adi',message:'hello adi', time:'25 january', profileUrl:'https://img-o.okeinfo.net/content/2020/04/  
20   ChartModel(name:'ashraf',message:'hello ashraf', time:'19.00', profileUrl:'https://img-o.okeinfo.net/content/2020/04/  
21   ChartModel(name:'niva',message:'hello niva', time:'12.21', profileUrl:'https://img-o.okeinfo.net/content/2020/04/23/  
22   ChartModel(name:'citra',message:'hello citra', time:'13.41', profileUrl:'https://img-o.okeinfo.net/content/2020/04/2  
23   ChartModel(name:'adi',message:'hello adi', time:'25 january', profileUrl:'https://img-o.okeinfo.net/content/2020/04/  
24   ChartModel(name:'ashraf',message:'hello ashraf', time:'19.00', profileUrl:'https://img-o.okeinfo.net/content/2020/04/  
25 ];
```

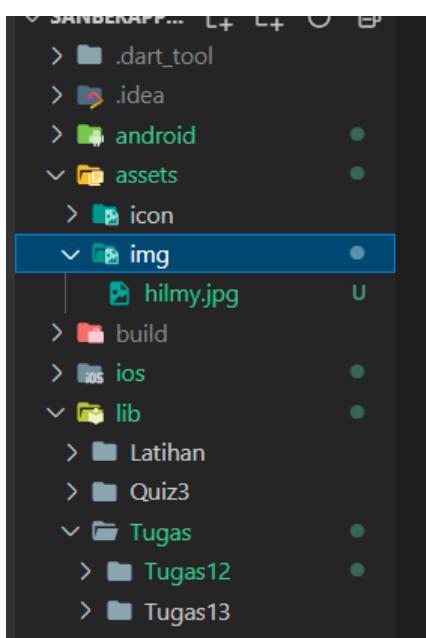
pubspec.yaml



```
Telegram.dart U  Chart_model.dart U ●  pubspec.yaml U X  ⌂ ⌂ ⌂ ⌂  
pubspec.yaml  
dev_dependencies:  
33 |   flutter_test:  
34 |     sdk: flutter  
35  
36 # For information on the generic Dart part of this file, see the  
37 # following page: https://dart.dev/tools/pub/pubspec  
38  
39 # The following section is specific to Flutter.  
40 flutter:  
41  
42 # The following line ensures that the Material Icons font is  
43 # included with your application, so that you can use the icons in  
44 # the material Icons class.  
45 uses-material-design: true  
46  
47 # To add assets to your application, add an assets section, like this:  
48 assets:  
49   - assets/img/  
50   #   - images/a_dot_ham.jpeg  
51  
52 # An image asset can refer to one or more resolution-specific "variants", see  
53 # https://flutter.dev/assets-and-images/#resolution-aware.  
54  
55 # For details regarding adding assets from package dependencies, see  
56 # https://flutter.dev/assets-and-images/#from-packages  
57
```

uncomment assets ubah jadi assets/img

assets/img



untuk foto silahkan menggunakan foto masing-masing.

f. Hasil dan Pembahasan

Dokumentasi tugas berupa laporan dalam mengimplementasikan widget core component pada flutter, dan juga memahami konsep widget core component.

g. Kesimpulan

Mahasiswa memahami penerapan widget core component pada flutter, dan juga memahami konsep widget core component.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|--------------------|--|-------|-----|-----|-----|
| 1 | Ketepatan waktu dan ketepatan dalam menjelaskan secara detail setiap baris code pada implementasi card view dan recycler view | 1 | 2 | 3 | (4) |
| 2 | Ketepatan waktu dan ketepatan dalam menjelaskan setiap baris code pada implementasi card view dan recycler view | 1 | 2 | (3) | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat dalam menjelaskan setiap baris code pada implementasi card view dan recycler view | 1 | (2) | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan dalam menjelaskan setiap baris code pada implementasi card view dan recycler view | (1) | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 17 : FLUTTER STYLING, NAVIGATION, & BUTTON TAB NAVIGATION

| | |
|----------------------------------|---|
| Pokok Bahasan | : Flutter Styling |
| Acara Praktikum/Pertemuan | : Minggu 5 / 1 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami konsep dalam flutter styling dan navigation.
2. Mahasiswa mampu menerapkan atau mengimplementasikan flutter styling dan navigation.

b. Indikator

Keberhasilan mahasiswa dalam memahami konsep dalam flutter styling dan navigation.

c. Dasar Teori

Flutter menggunakan material design sebagai style default untuk mengatur UI-nya, meskipun kita diberikan kebebasan untuk memodifikasi sesuai dengan keinginan. Maka pada bagian pertama ini, kita akan belajar bagaimana menggunakan widget Scaffold, serta attribute apa saja yang perlu kita ketahui.

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

Memulai dari awal untuk mengenal satu persatu widget Flutter, dimulai dari duet maut MaterialApp & Scaffold, Buka file lib/main.dart, lalu menjadi

```
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return null;
}
```

Penjelasan: Semua widget yang biasanya digunakan beserta style-nya akan dirangkum oleh material.dart yang kita import pada line pertama. Ketika Flutter dijalankan, maka secara otomatis dia akan mengeksekusi class yang menjadi value dari runApp() yang berada di dalam method main(). Class MyApp() meng-extend StatelessWidget, dimana Flutter memiliki dua buah widget yang bernama Stateless dan StatefulWidget. Perbedaan keduanya secara detail tidak akan kita bahas pada kesempatan kali ini, singkatnya StatelessWidget adalah widget yang tidak akan berubah setelah diinisiasi atau setelah dimuat. Sebaliknya berlaku untuk StatefulWidget adalah widget yang bisa berubah ketika mendapatkan perintah dari pengguna.

Tiba saatnya untuk menggunakan duet **MaterialApp** widget dan **Scaffold** widget, di dalam file yang sama, modifikasi menjadi

```
import 'package:flutter/material.dart';

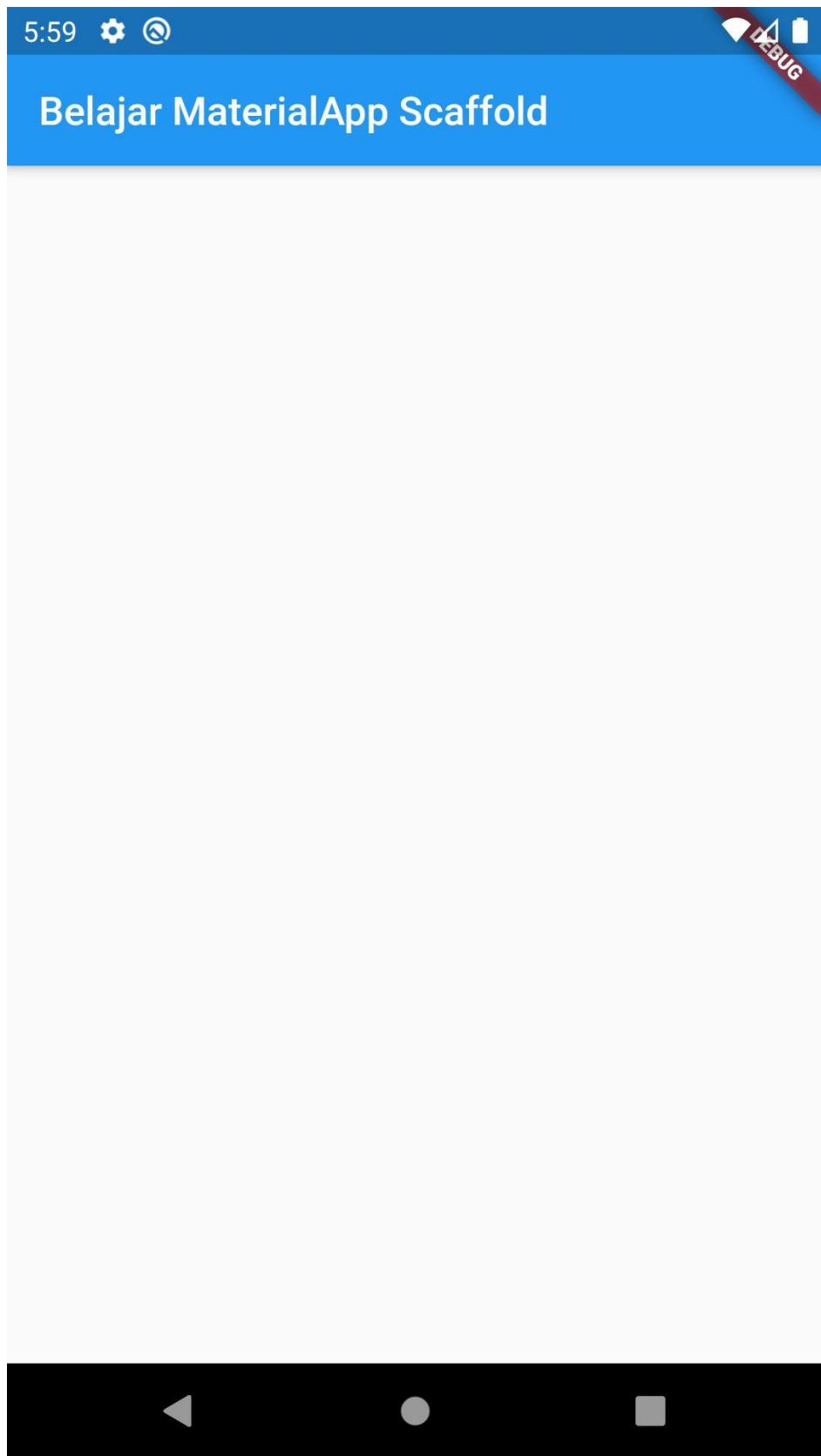
void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: Text("Belajar MaterialApp Scaffold")),
      ),
    );
  }
}
```

Penjelasan: Ada beberapa bagian yang perlu dijelaskan, diantaranya MaterialApp adalah sebuah parent dimana yang diapitnya akan menerapkan style material design. Selain itu, dia juga memiliki control untuk mengatur route, theme, supported locales, dan lain sebagainya. Akan tetapi pada seri kali ini kita hanya akan membahas tentang theme-nya saja, adapun fungsi lainnya akan dirangkaikan dengan materi lainnya.

Scaffold memiliki peran untuk mengatur struktur visual layout dengan mengimplementasikan material design, dimana dia juga memiliki kemampuan untuk membuat app bars, drawers, snack bars, bottom sheets dan lain sebagainya.

appBar adalah salah satu properti yang dimiliki oleh **Scaffold** widget untuk membuat sebuah bar pada aplikasi, salah satu contohnya adalah menampilkan teks Belajar **MaterialApp Scaffold**. Adapun hasil yang akan kita peroleh seperti gambar di bawah ini



Menyelami lebih dalam lagi, kita akan menerapkan beberapa bagian dari attribute appBar().

Dengan file yang sama, modifikasi menjadi

```
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          leading: Icon(Icons.dashboard),
          title: Text("Belajar MaterialApp Scaffold"),
          actions: <Widget>[
            Icon(Icons.search),
            // Icon(Icons.find_in_page)
          ],
          actionsIconTheme: IconThemeData(color: Colors.yellow),
          backgroundColor: Colors.pinkAccent,
          bottom: PreferredSize(
            child: Container(
              color: Colors.orange,
              height: 4.0,
            ),
            preferredSize: Size.fromHeight(4.0)
          ),
          centerTitle: true,
        ),
        debugShowCheckedModeBanner: false,
      );
  }
}
```

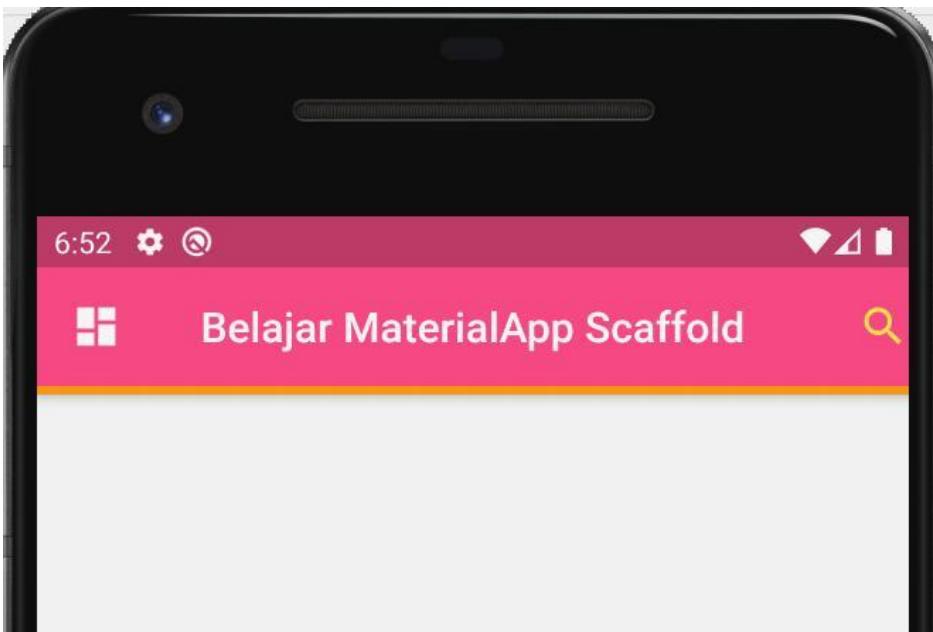
Penjelasan: Berikut beberapa penjelasan dari penggunaan attribute di atas, diantaranya **leading**, **title** dan **actions** adalah kesatuan yang membentuk urutan penempatan sesuai dengan urutan nama yang saya bold diawal. Ketiganya bisa menampilkan apa saja yang diinginkan, adapun leading dan title hanya bisa memuat satu widget, sedangkan actions bisa menampung lebih dari satu widget sesuai kebutuhan.

actionsIconTheme akan mengatur theme yang ingin diterapkan pada actions dengan catatan jika yang dimuat adalah widget Icon.

bottom sesuai namanya akan ditempatkan dibawah appbar, misalnya saja kita ingin memberikan efek garis pada bottom appbar.

debugShowCheckedModeBanner secara default bernilai **true** yang akan menampilkan debug banner pada pojok kanan atas, set value-nya jadi **false** untuk menghilangkan debug banner tersebut.

Adapun hasil yang akan kita peroleh seperti gambar dibawah ini



Lebih jauh lagi jika diteruskan, **Scaffold** memiliki kemampuan untuk membuat drawers (menu yang ada disamping), snackbars dan lain sebagainya. Akan tetapi pada bagian ini tidak akan dibahas, maka perubahan terakhir yang akan dilakukan adalah mengenal 2 attribute lainnya yakni floatingActionButton dan body, modifikasi file yang sama menjadi.

```
import 'package:flutter/material.dart';

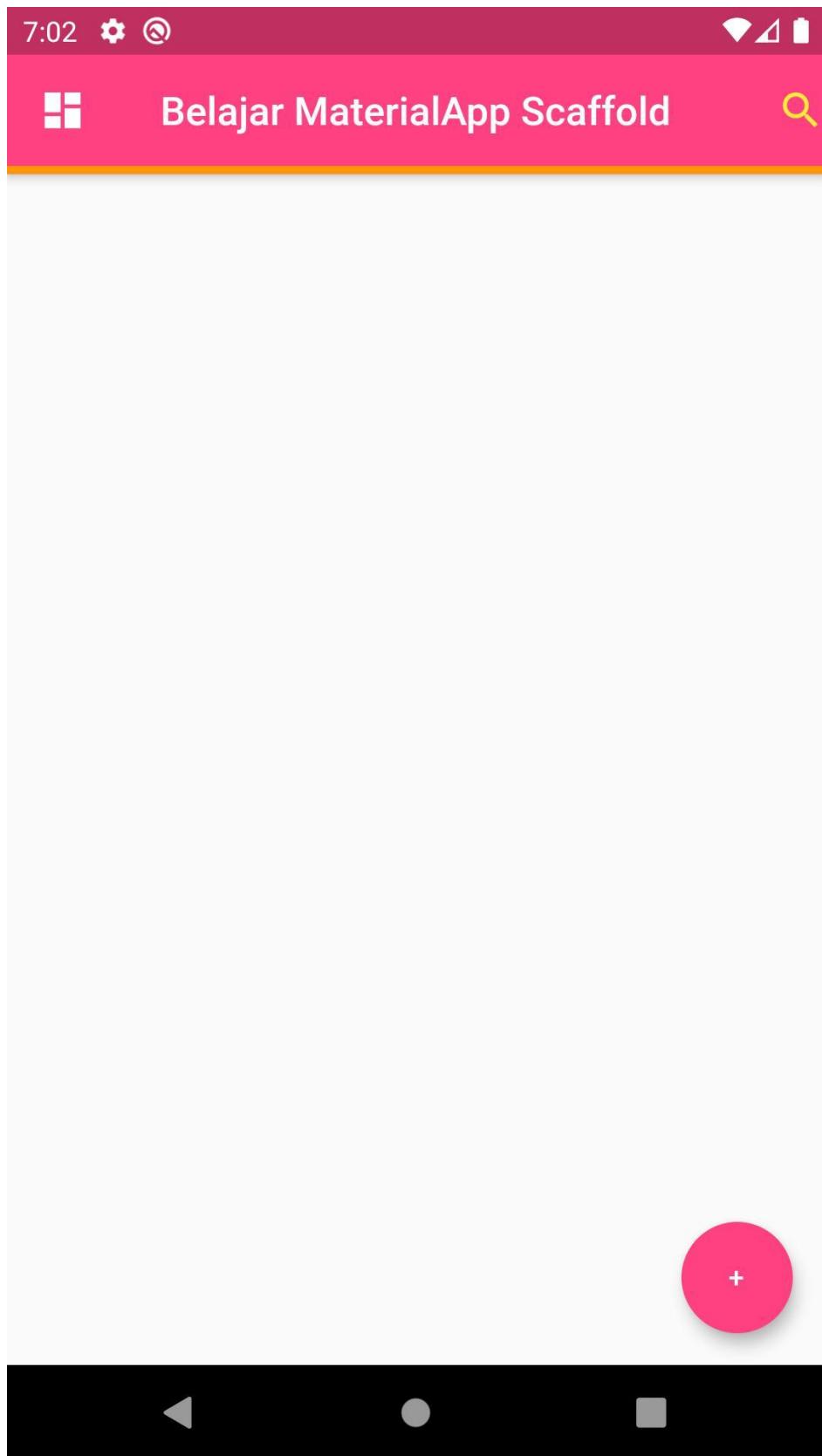
void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          leading: Icon(Icons.dashboard),
          title: Text("Belajar MaterialApp Scaffold"),
          actions: <Widget>[
            Icon(Icons.search),
            // Icon(Icons.find_in_page)
          ],
          actionsIconTheme: IconThemeData(color: Colors.yellow),
          backgroundColor: Colors.pinkAccent,
          bottom: PreferredSize(
            child: Container(
              color: Colors.orange,
              height: 4.0,
            ),
            preferredSize: Size.fromHeight(4.0)
          ),
          centerTitle: true,
        ),
      ),
    );
  }
}
```

```
) ,  
  
    //PERUBAHAN BARU  
    floatingActionButton: FloatingActionButton(  
        backgroundColor: Colors.pinkAccent,  
        child: Text('+'),  
        onPressed: () {},  
    ),  
    body: null,  
    //PERUBAHAN BARU  
),  
debugShowCheckedModeBanner: false,  
);  
}  
}
```

Penjelasan: floatingActionButton digunakan untuk membuat tombol melayang pada pojok kanan bawah, sedangkan body memuat content yang akan ditampilkan pada frame putih yang sedang kosong saat ini.

Adapun tampilan akhir yang akan kita peroleh terlihat seperti gambar di bawah ini.



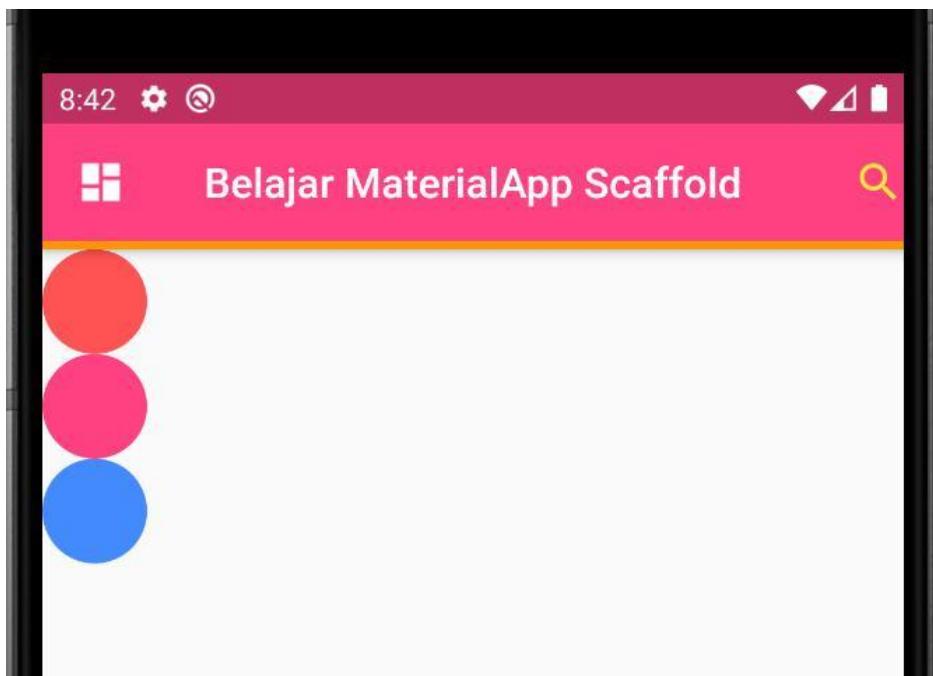
Pasangan Row-Column

Jika anda berlatar belakang dari classic web developer, maka menggambar row-column bisa diasumsikan sebagai table yang memiliki row dan column. Sebab dahulu kala, mengatur tata letak bisa memanfaatkan tag table pada html. Jadi Row untuk mengatur widget agar tersusun secara horizontal (red: dari kiri ke kanan), sedangkan **Column** berlaku sebaliknya, yakni secara vertical (red: dari atas ke bawah).

Pertama, kita akan belajar bagaimana cara kerja dari Column dengan schema yang diinginkan adalah menampilkan 3 buah lingkaran yang tersusun ke bawah. Sekedar informasi bahwa kita hanya akan fokus pada bagian content saja, maka seluruh code-nya tidak akan saya tampilkan lagi, melainkan hanya pada bagian attribute body dari **Scaffold**. Buka file lib/main.dart dan masukkan code berikut pada bagian body.

```
body: Column(children: <Widget>[  
    Container(width: 50, height: 50, decoration: BoxDecoration(color: Colors.redAccent,  
shape: BoxShape.circle),),  
    Container(width: 50, height: 50, decoration: BoxDecoration(color: Colors.pinkAccent,  
shape: BoxShape.circle),),  
    Container(width: 50, height: 50, decoration: BoxDecoration(color: Colors.blueAccent,  
shape: BoxShape.circle),),  
],),
```

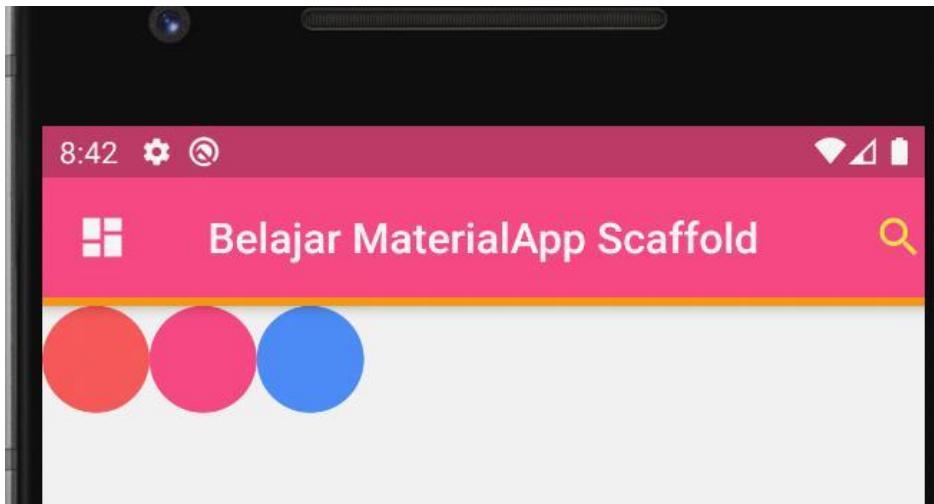
Penjelasan: Column bisa menampung lebih dari 1 widget, sehingga kita bisa memasukkan widget lainnya di dalam children-nya sesuka hati. Seperti yang disinggung di awal bahwa column akan menyusun widget yang ada di dalamnya dari atas ke bawah.



Penerapan Row sama saja dengan Column, bisa menampung lebih dari 1 widget di dalamnya.

Hanya saja berbeda orientasi penggunaannya saja. Buka file yang sama dan modifikasi menjadi.

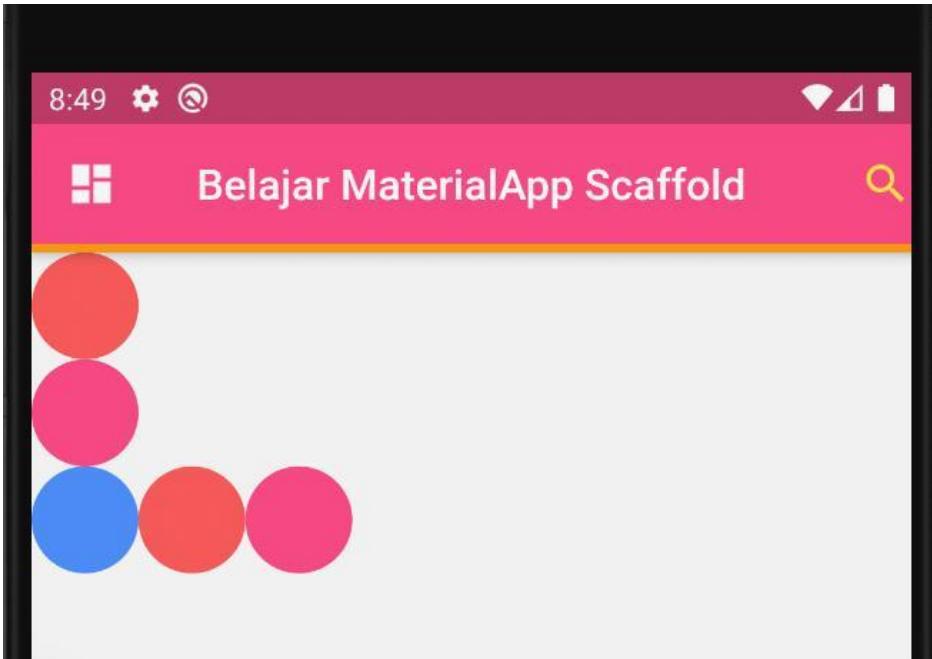
```
body: Row(  
    children: <Widget>[  
        Container(width: 50, height: 50, decoration: BoxDecoration(color: Colors.redAccent,  
shape: BoxShape.circle),),  
        Container(width: 50, height: 50, decoration: BoxDecoration(color: Colors.pinkAccent,  
shape: BoxShape.circle),),  
        Container(width: 50, height: 50, decoration: BoxDecoration(color: Colors.blueAccent,  
shape: BoxShape.circle),),  
    ],  
) ,
```



Selain bekerja secara individu, kita juga bisa memadukan antara row dan column dalam satu keadaan. Misalnya saja case yang diinginkan adalah 3 lingkaran membentuk urutan kebawah dan pada lingkaran terakhir urutannya ke samping.

```
body: Column(  
    mainAxisAlignment: MainAxisAlignment.start,  
    children: <Widget>[  
        Container(width: 50, height: 50, decoration: BoxDecoration(color: Colors.redAccent,  
shape: BoxShape.circle),),  
        Container(width: 50, height: 50, decoration: BoxDecoration(color: Colors.pinkAccent,  
shape: BoxShape.circle),),  
        Row(  
            children: <Widget>[  
                Container(width: 50, height: 50, decoration: BoxDecoration(color:  
Colors.blueAccent, shape: BoxShape.circle),),  
                Container(width: 50, height: 50, decoration: BoxDecoration(color: Colors.redAccent,  
shape: BoxShape.circle),),  
                Container(width: 50, height: 50, decoration: BoxDecoration(color:  
Colors.pinkAccent, shape: BoxShape.circle),),  
            ],  
        ),  
    ],  
) ,
```

Penjelasan: Perpaduan antara **Row & Column** di Flutter, dimana Row bisa dimasukkan ke dalam Column, begitupun sebaliknya. Ada hal baru yang kita pelajari yakni penggunaan crossAxisAlignment, dimana attribute ini berfungsi untuk mengatur alignment dari Row maupun Column. Adapun sifatnya selalu berlawanan dengan orientasi dari Row & Column tersebut. Misalnya saja, Column memiliki sifat penerapan secara vertical, maka crossAxisAlignment akan mengurnya secara horizontal, begitupun dengan row.



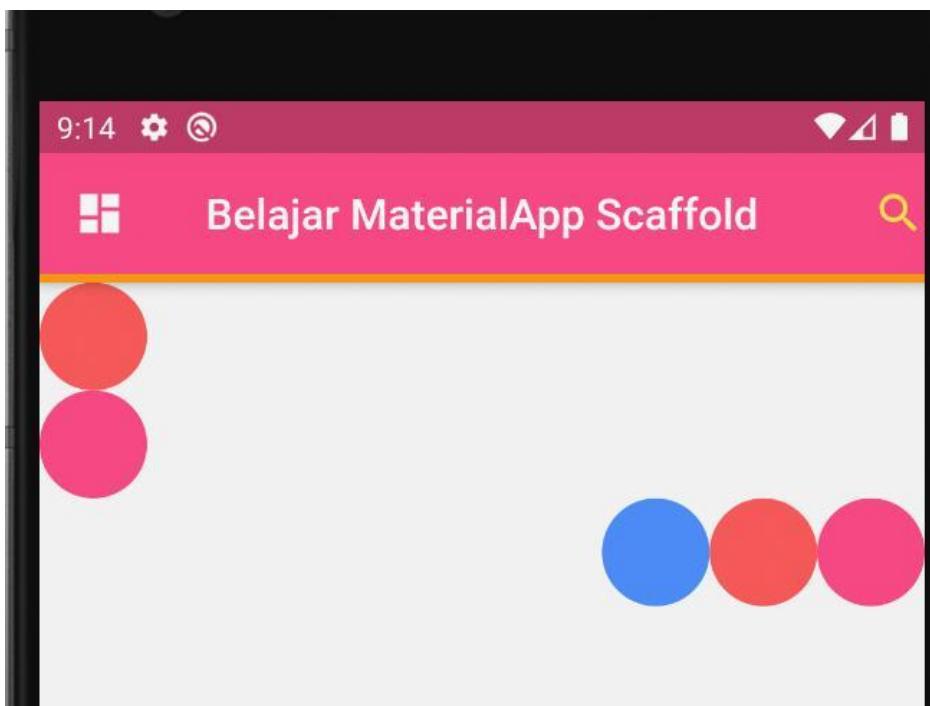
Selain crossAxisAlignment, ada attribute lain yang sama pentingnya untuk memaksimalkan fungsi Row & Column. Misalnya saja schema yang diinginkan adalah kita ingin mengatur 3 lingkaran paling bawah akan disimpan pada sudut kanan screen, sedangkan dua lingkaran lainnya tetap pada posisinya.

Pada schema diatas, kita akan memanfaatkan peran mainAxisAlignment, tambahkan code berikut sebagai attribute row

```
body: Column(
  crossAxisAlignment: CrossAxisAlignment.start,
  children: <Widget>[
    Container(width: 50, height: 50, decoration: BoxDecoration(color: Colors.redAccent,
shape: BoxShape.circle),),
    Container(width: 50, height: 50, decoration: BoxDecoration(color: Colors.pinkAccent,
shape: BoxShape.circle),),
    Row(
      //TAMBAHKAN CODE INI
      mainAxisAlignment: MainAxisAlignment.end,
      //TAMBAHKAN CODE INI
```

```
children: <Widget>[
    Container(width: 50, height: 50, decoration: BoxDecoration(color:
Colors.blueAccent, shape: BoxShape.circle),),
    Container(width: 50, height: 50, decoration: BoxDecoration(color: Colors.redAccent,
shape: BoxShape.circle),),
    Container(width: 50, height: 50, decoration: BoxDecoration(color:
Colors.pinkAccent, shape: BoxShape.circle),),
],
),
],
),
),
```

Penjelasan: mainAxisAlignment memiliki peran yang sama dengan crossAxisAlignment, yakni untuk mengatur aligment dari widget yang ada di dalamnya. Hanya saja mainAxisAlignment memiliki sifat yang sejalan dengan widget-nya. Misalnya **mainAxisAlignment** pada Row akan mengatur aligment secara horizontal, sehingga jika value-nya adalah .end maka widget-nya akan ditempatkan diakhir screen secara horizontal.



Mengatur Padding

Padding adalah jarak atau ruang yang berada di antara border dan konten. Sedangkan margin adalah jarak atau ruang yang berada atau dihitung dari border ke arah luar atau jarak antara border dan konten yang berada di luar border.

Cara pertama kita bisa menggunakan Widget Padding, dimana widget ini akan mengatur padding untuk widget yang ada di dalamnya.

```
Padding(  
  padding: EdgeInsets.all(8.0),  
  child: const Card(child: Text('Hello World!')),  
)
```

Pada contoh di atas akan menghasilkan padding di semua sisi yaitu atas, kanan, bawah dan kiri. Contoh lain bisa seperti berikut.

```
Padding(  
  padding: EdgeInsets.fromLTRB(8.0, 5.0, 5.0, 4.0),  
  child: const Card(child: Text('Hello World!')),  
)
```

Pada contoh di atas akan menghasilkan padding di semua sisi dengan urutan kiri (L), atas (T), kanan (R), bawah (B).

Contoh lain untuk mengatur pada sisi tertentu saja bisa seperti berikut.

```
Padding(  
  padding: EdgeInsets.only(left:8.0, right:8.0),  
  child: const Card(child: Text('Hello World!')),  
)
```

Pada contoh di atas akan menghasilkan padding hanya pada sisi yang kita sebutkan saja. Seperti pada contoh saya hanya menyebut left dan right. Selain menggunakan widget Padding kita juga bisa menggunakan Container sebagai berikut.

```
Container(  
  padding: EdgeInsets.only(left:8.0, right:8.0),  
  child: const Card(child: Text('Hello World!')),  
)
```

Mengatur Margin

Untuk margin kita bisa menggunakannya pada widget container dikarenakan tidak ada widget khusus untuk margin seperti yang dimiliki padding. Sehingga penggunaannya akan seperti berikut.

```
Container(  
  margin: EdgeInsets.all(8.0),  
  child: const Card(child: Text('Hello World!')),  
)
```

atau materi video dapat di rangkum di video ini

<https://www.youtube.com/watch?v=PFkCSLnKKx0>

<https://www.youtube.com/watch?v=PFkCSLnKKx0>

<https://www.youtube.com/watch?v=PFkCSLnKKx0>

<https://www.youtube.com/watch?v=PFkCSLnKKx0>

f. Hasil dan Pembahasan

Dokumentasi tugas berupa laporan dalam mengimplementasikan flutter styling dan navigation.

g. Kesimpulan

Mahasiswa memahami konsep flutter styling dan navigation dengan menerapkan atau mengimplementasikan flutter styling dan navigation.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|--------------------|---|-------|-----|-----|-----|
| 1 | Ketepatan waktu dan ketepatan dalam menjelaskan secara detail tugas review video | 1 | 2 | 3 | (4) |
| 2 | Ketepatan waktu dan ketepatan dalam menjelaskan dari tugas review video | 1 | 2 | (3) | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat dalam menjelaskan tugas review video | 1 | (2) | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan dalam menjelaskan tugas review video | (1) | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 18 : FLUTTER STYLING, NAVIGATION, & BUTTON TAB NAVIGATION

| | |
|----------------------------------|---|
| Pokok Bahasan | : Flutter Navigation |
| Acara Praktikum/Pertemuan | : Minggu 5 / 2 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami konsep dalam flutter styling dan navigation.
2. Mahasiswa mampu menerapkan atau mengimplementasikan flutter styling dan navigation.

b. Indikator

Keberhasilan mahasiswa dalam memahami flutter styling dan navigation, mengimplementasikan flutter styling dan navigation.

c. Dasar Teori

Pada Materi kali ini kita akan membahas navigasi routing pada flutter. dalam sebuah aplikasi mobile biasanya memiliki full-screen elemen yang disebut "screen" atau "page". di Flutter element tersebut disebut route dan di kelola oleh widget Navigator. Widget Navigator berfungsi untuk menampilkan konten ke halaman atau layar baru. jika pada native android, Navigator route di namakan activity dan di ios sebagai viewController. wodget Navigator bekerja seperti tumpukan layar(stack), ia menggunakan prinsip LIFO(Last-In, First Out). ada 2 method yang dapat digunakan untuk navigtor pada widget yaitu:

1. navigator.push() : Methode push digunakan untuk menambahkan rute lain keatas tumpukan screen(stack)saat ini. Halaman baru ditampilkan diatas halaman sebelumnya.
2. Navigator.pop(): Methode pop menghapus rute paling atas dari tumpukan. ini menampilkan halaman sebelumnya kepada pengguna. Simple Routing Flutter Penggunaan widget Navigator untuk routing di flutter yaitu seperti dibawah ini

```
Navigator.push(  
  context,  
  MaterialPageRoute(builder: (context) {  
    return AboutPage()  
  })  
) ;
```

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

Dalam contoh penerapannya, kita akan membuat routing sederhana menggunakan metode Navigator.push untuk navigasi ke layar baru dan Navigator.pop untuk kembali ke layar sebelumnya.

main.dart

```
import 'package:flutter/material.dart';
void main() {
  runApp(MaterialApp(
    home: HomePage(),
  ));
}
class HomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Belajar Routing'),
      ),
      body: Center(
        child: ElevatedButton(
          onPressed: () {
            Route route = MaterialPageRoute(builder: (context) => AboutPage());
            Navigator.push(context, route);
          },
          child: Text('Tap Untuk ke AboutPage'),
        ),
      ),
    );
  }
}
class AboutPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Tentang Aplikasi'),
      ),
      body: Center(
        child: ElevatedButton(
          onPressed: () {
            Navigator.pop(context);
          },
          child: Text('Kembali'),
        ),
      ),
    );
}
```

```
}
```

Contoh kode di atas membuat dua stateless widget dimana halaman awal menggunakan HomePage dan berpindah ke AboutPage saat tombol di tap. Tampilannya seperti gambar di bawah ini



f. Hasil dan Pembahasan

Dokumentasi tugas berupa laporan dalam mengimplementasikan flutter styling dan navigation.

g. Kesimpulan

Mahasiswa memahami konsep flutter styling dan navigation dengan menerapkan atau mengimplementasikan flutter styling dan navigation.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|--------------------|--|-------|-----|-----|-----|
| | | 1 | 2 | 3 | 4 |
| 1 | Ketepatan waktu dan ketepatan implementasi pada percobaan fragment dengan penjelasan singkat | | | | (4) |
| 2 | Ketepatan waktu dan ketepatan implementasi pada percobaan fragment | 1 | 2 | (3) | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat implementasi pada percobaan fragment | 1 | (2) | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan implementasi pada percobaan fragment | (1) | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 19 : FLUTTER STYLING, NAVIGATION, & BUTTON TAB NAVIGATION

| | |
|----------------------------------|---|
| Pokok Bahasan | : Button Tab Navigation |
| Acara Praktikum/Pertemuan | : Minggu 5 / 3 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami konsep dalam navigation pada flutter.
2. Mahasiswa mampu menerapkan atau mengimplementasikan navigation pada flutter.

b. Indikator

Keberhasilan mahasiswa dalam memahami konsep flutter styling dan navigation, implementasi fragment pada android, serta analisa source code.

c. Dasar Teori

Named Routing

Sesuai namanya, Named Routing yaitu memberi nama pada routing dengan tujuan untuk mempermudah dalam membaca dan menentukan arah dari suatu navigasi. Sedikit berbeda dengan simple routing, disini kita akan menggunakan Navigator.pushNamed untuk menuju ke halaman baru namun tetap menggunakan Navigator.pop untuk kembali ke halaman sebelumnya.

Route Generator

Meskipun named routing terlihat lebih baik dari simple routing, namun apabila ingin membuat aplikasi dengan skala yang cukup besar atau ingin mengatur animasi transisi dari tiap routing maka onGenerateRoute merupakan pilihan yang tepat. Dalam tutorial route generator ini kita akan coba memisahkan antara main, screen dan routing file

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

Named Routing

`Navigator.pushNamed` membutuhkan dua properti wajib yaitu context dan string sebagai nama routenya. Kita juga dapat mengirim parameter object ke dalam route.

```
Navigator.pushNamed(context, String, {Object});  
import 'package:flutter/material.dart';  
void main() {  
  runApp(MaterialApp(  
    initialRoute: '/',  
    routes: <String, WidgetBuilder>{  
      '/': (context) => HomePage(),  
      '/about': (context) => AboutPage(),  
    },  
  ));  
}  
class HomePage extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: Text('Belajar Routing'),  
      ),  
      body: Center(  
        child: ElevatedButton(  
          onPressed: () {  
            Navigator.pushNamed(context, '/about');  
          },  
          child: Text('Tap Untuk ke AboutPage'),  
        ),  
      ),  
    );  
  }  
}  
class AboutPage extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: Text('Tentang Aplikasi'),  
      ),  
      body: Center(  
        child: ElevatedButton(  
          onPressed: () {  
            Navigator.pop(context);  
          },  
          child: Text('Kembali'),  
        ),  
      ),  
    );  
  }  
}
```

Contoh kode diatas akan sama dengan gambar 1.1.

Perhatikan potongan kode pada main function dibawah ini

```
void main() {  
  runApp(MaterialApp(  
    initialRoute: '/',
```

```

        routes: <String, WidgetBuilder>{
          '/': (context) => HomePage(),
          '/about': (context) => AboutPage(),
        },
      ) );
}

```

method initialRoute tidak wajib jika dalam routes kita memiliki “/”. Kita juga dapat menggunakan properti home sebagai initialRoute seperti di bawah ini.

```

void main() {
  runApp(MaterialApp(
    home: HomePage(),
    routes: <String, WidgetBuilder>{
      '/': (context) => HomePage(),
      '/about': (context) => AboutPage(),
    },
  ) );
}

```

Route Generator

```

Main.dart
import 'package:flutter/material.dart';
import 'package:belajar_flutter/routes.dart';
void main() {
  runApp(MaterialApp(
    onGenerateRoute: RouteGenerator.generateRoute,
  ) );
}

screen.dart
import 'package:flutter/material.dart';
class HomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Belajar Routing'),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            ElevatedButton(
              onPressed: () {
                Navigator.pushNamed(context, '/about');
              },
              child: Text('Tap Untuk ke AboutPage'),
            ),
            ElevatedButton(
              onPressed: () {
                Navigator.pushNamed(context, '/halaman-404');
              },
              child: Text('Tap Halaman lain'),
            ),
          ],
        ),
      ),
    );
}

```

```

        ),
    );
}
}
class AboutPage extends StatelessWidget {
@override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: Text('Tentang Aplikasi'),
        ),
        body: Center(
            child: ElevatedButton(
                onPressed: () {
                    Navigator.pop(context);
                },
                child: Text('Kembali'),
            ),
        ),
    );
}
}

```

routes.dart

```

import 'package:flutter/material.dart';
import 'package:belajar_flutter/screen.dart';
class RouteGenerator {
    static Route<dynamic> generateRoute(RouteSettings settings) {
        // jika ingin mengirim argument
        // final args = settings.arguments;
        switch (settings.name) {
            case '/':
                return MaterialPageRoute(builder: (_) => HomePage());
            case '/about':
                return MaterialPageRoute(builder: (_) => AboutPage());
                // return MaterialPageRoute(builder: (_) => AboutPage(args));
            default:
                return _errorRoute();
        }
    }
    static Route<dynamic> _errorRoute() {
        return MaterialPageRoute(builder: (_) {
            return Scaffold(
                appBar: AppBar(title: Text("Error")),
                body: Center(child: Text('Error page')),
            );
        });
    }
}

```

pada contoh diatas kita juga menambahkan error page dimana jika routing tidak ditemukan maka akan menampilkan halaman error tersebut



f. Hasil dan Pembahasan

Project aplikasi dari percobaan implementasi code fragment.

g. Kesimpulan

Mahasiswa berhasil meng-implementasi-kan fragment pada project individu serta memahami konsep implementasi fragment.

a. Rubrik Penilaian

| No | Indikator | Skor* |
|----|-----------|-------|
|----|-----------|-------|

| | | | | | |
|--------------------|--|---|---|---|---|
| 1 | Ketepatan waktu dan ketepatan implementasi pada percobaan fragment dengan penjelasan singkat | 1 | 2 | 3 | 4 |
| 2 | Ketepatan waktu dan ketepatan implementasi pada percobaan fragment | 1 | 2 | 3 | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat implementasi pada percobaan fragment | 1 | 2 | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan implementasi pada percobaan fragment | 1 | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 20 : FLUTTER STYLING, NAVIGATION, & BUTTON TAB NAVIGATION

| | |
|----------------------------------|---|
| Pokok Bahasan | : Drawer Widget |
| Acara Praktikum/Pertemuan | : Minggu 5 / 4 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami konsep dalam button navigation dengan drawer widget pada flutter.
2. Mahasiswa mampu menerapkan atau mengimplementasikan button navigation dengan drawer widget pada flutter.

b. Indikator

Keberhasilan mahasiswa dalam memahami konsep button navigation dengan drawer widget, implementasi button navigation dengan drawer widget pada flutter.

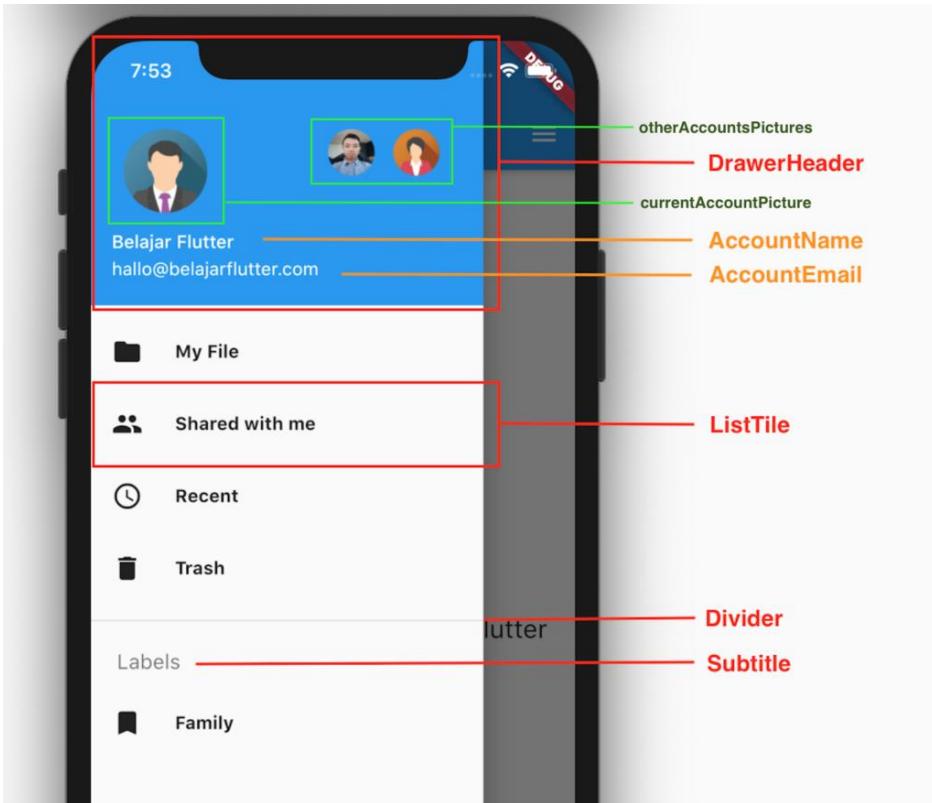
c. Dasar Teori

Sebelum ke tahap cara membuat navigation drawer, ada baiknya kita mengerti tentang Drawer itu sendiri. Drawer widget merupakan single child widget yang artinya hanya dapat memiliki satu child widget di dalamnya. Karena Drawer hanya memiliki properti child dan bukan children, maka untuk menampatkan item-item lain bisa menggunakan ListView widget. Sebenarnya tidak ada keharusan menggunakan ListView namun keuntungan menggunakan ListView widget dibandingkan dengan column atau widget lainnya yaitu untuk memudahkan dalam mengatur list Item dan vertical scrolling apabila item menu melebihi tinggi layar.

Anatomi Drawer

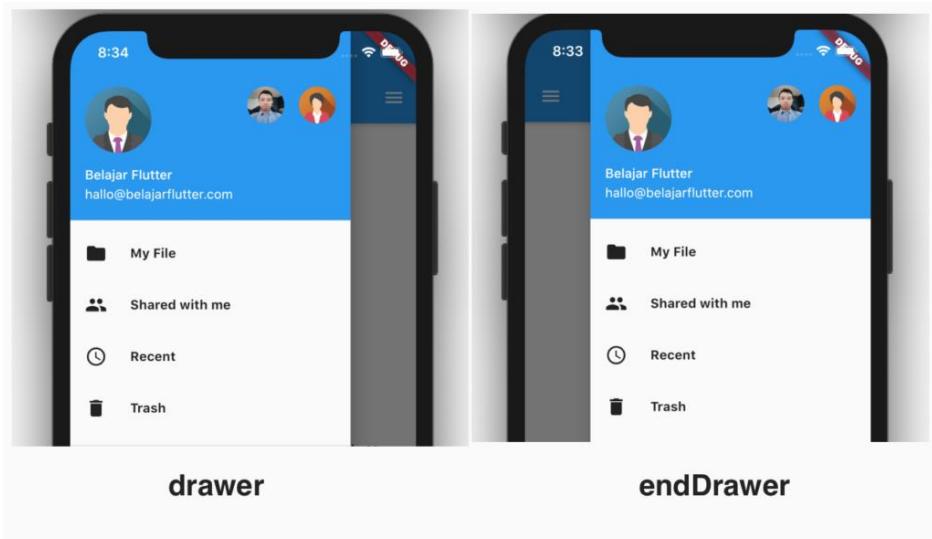
Anatomi dari sebuah navigation drawer menurut material design itu ada delapan (8) point, namun disini kita sederhanakan saja menjadi 4 bagian yaitu

1. Header,
2. List Item,
3. Divider dan
4. Subtitle



Jenis Drawer

Jenis Drawer navigasi pada flutter dibagi menjadi dua sesuai dengan letak posisi drawer itu sendiri yaitu drawer dan endDrawer. Drawer yang muncul di sebelah kiri dinamakan drawer, sedangkan apabila muncul dari sebelah kanan dinamakan endDrawer karena menggunakan properti endDrawer



Cara membuat navigation drawer pada flutter :

<https://www.youtube.com/watch?v=Rp3RzZL5VaQ>

Bottom Tab Navigator

untuk dokumentasi lengkap dari bottom tab :

<https://api.flutter.dev/flutter/material/BottomNavigationBar-class.html>

Bottom Tab



Berikut cara pembuatannya

<https://www.youtube.com/watch?v=1y-gfskpMIM>

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

gabungkan 3 navigator yang ada pada flutter yang pertama :

1. berpindah halaman dari halaman lainnya yang telah dibuat pada tugas sebelumnya ke halaman berikutnya.
2. buat bottom tab navigator untuk halaman homescreen, search dan akun
3. buat drawer navigator untuk menampilkan drawer yang ada pada samping kiri screen

f. Hasil dan Pembahasan

Dokumentasi tugas berupa laporan secara individu penjelasan setiap code pada project fragment yang di buktikan dengan screenshot pada proses percobaan.

g. Kesimpulan

Mahasiswa memahami konsep dasar fragment dengan mampu memahami setiap baris code dari implementasi fragment.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|--------------------|--|-------|-----|-----|-----|
| 1 | Ketepatan waktu dan ketepatan dalam menjelaskan secara detail setiap baris code pada implemenasi fragment | 1 | 2 | 3 | (4) |
| 2 | Ketepatan waktu dan ketepatan dalam menjelaskan setiap baris code pada implemenasi fragment | 1 | 2 | (3) | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat dalam menjelaskan setiap baris code pada implemenasi fragment | 1 | (2) | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan dalam menjelaskan setiap baris code pada implemenasi fragment | (1) | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 21 : GRADIENT, PAGEVIEW, DROPDOWN

Pokok Bahasan

: Gradient

Acara Praktikum/Pertemuan

: Minggu 6 / 1

Tempat

: Daring / Luring Politeknik Negeri Jember

Alokasi Waktu

: 100 menit

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami konsep Gradient, Pageview, Dan Dropdown Menu pada flutter
2. Mahasiswa mampu menerapkan Gradient, Pageview, Dan Dropdown Menu pada flutter

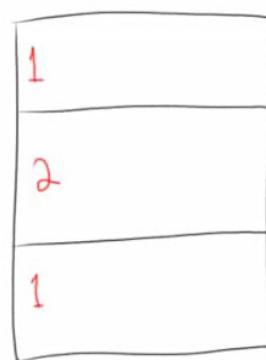
b. Indikator

Kemampuan mahasiswa dalam memahami konsep Gradient, Pageview, Dan Dropdown Menu pada flutter

c. Dasar Teori

• Gradient

1. Gradient merupakan corak warna yang sanggup dikembangkan dari dua warna atau lebih, atau perpaduan antara warna yang digabungkan. Untuk menciptakan efek gradient pada Flutter sangat mudah, bahkan ada beberapa tipe gradient.
2. Menciptakan background dengan efek gradient, berikut contoh tampilan yang akan dibuat untuk menampilkan background dengan efek gradient :



3. Membagi layar menjadi 3 bagian, dengan memakai widget **Flexible** dan masing-masing perbandingannya yakni 1:2:1. Layout tersebut akan kita gunakan untuk menciptakan efek gradient yang berbeda-beda, yaitu **LinearGradient**, **RadialGradient**, dan **SweepGradient**.

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

• **Gradient**

1. Membuat project baru dengan nama minggu 7
2. Selanjutnya buka file **main.dart** pada folder **lib**, lalu hapus semua isi kodennya. Pertama tambahkan isyarat untuk import library material :

```
1 import 'package:flutter/material.dart';
```

3. Membuat fungsi main untuk memanggil class yang pertama kali ditampilkan, tambahkan isyarat berikut dibawah import :

```
3 void main() {  
4   runApp(new MaterialApp(  
5     |   home: new Home(),  
6     |   )); // MaterialApp  
7 }
```

4. Kode diatas class yang pertama kali dipanggil yaitu **Home**, selanjutnya membuat class Home berupa **StatefulWidget**. Menambahkan fungsi **Home** pada main.dart, seperti berikut :

```
9 class Home extends StatefulWidget {  
10   // This widget is the root of your application.  
11   @override  
12   _HomeState createState() => _HomeState();  
13 }
```

5. Menambahkan fungsi **_HomeState** seperti berikut :

```

15 < class _HomeState extends State<Home> {
16   @override
17   Widget build(BuildContext context) {
18     return new Scaffold(
19       body: new Container(
20         decoration: new BoxDecoration(
21           gradient: new LinearGradient(
22             begin: FractionalOffset.topCenter,
23             end: FractionalOffset.bottomCenter,
24             colors: [
25               Colors.white,
26               Colors.purpleAccent,
27               Colors.deepPurple
28             ]), // LinearGradient // BoxDecoration
29         ), // Container
30       ); // Scaffold
31     }
32   }

```

6. Jalankan dan analisa program

f. Hasil dan Pembahasan

1. Dokumentasi tugas berupa laporan (definisi, konsep dasar gradien, pageview, dan down menu).
2. Dokumentasi tugas berupa file pdf.

g. Kesimpulan

Mahasiswa memahami mengenai definisi git, konsep dasar pengenalan dan workflow flutter.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|--------------------|---|-------|-----|-----|-----|
| 1 | Ketepatan waktu dan ketepatan dalam menjelaskan dari tugas ditunjang dengan bukti referensi | 1 | 2 | 3 | (4) |
| 2 | Ketepatan waktu dan ketepatan dalam menjelaskan dari tugas | 1 | 2 | (3) | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat dalam menjelaskan tugas | 1 | (2) | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan dalam menjelaskan tugas | (1) | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 22 : GRADIENT, PAGEVIEW, DROPODOWN

| | |
|----------------------------------|---|
| Pokok Bahasan | : Pageview |
| Acara Praktikum/Pertemuan | : Minggu 6 / 2 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami konsep Gradient, Pageview, Dan Dropdown Menu pada flutter
2. Mahasiswa mampu menerapkan Gradient, Pageview, Dan Dropdown Menu pada flutter

b. Indikator

Kemampuan mahasiswa dalam memahami konsep Gradient, Pageview, Dan Dropdown Menu pada flutter

c. Dasar Teori

• **PageView**

PageView adalah widget Flutter yang membuat halaman yang dapat digulir di layar. Itu dapat membuat daftar sejumlah halaman tetap atau dapat menggunakan fungsi pembangun untuk membuat halaman sesuai permintaan. Widget PageView sangat mirip dengan widget ListView dalam hal membangun elemen. Apa saja konstruktor PageView yang berbeda?

1. PageView() – Membuat daftar halaman yang dapat digulir dari daftar widget eksplisit
2. PageView.builder() – Membuat daftar halaman yang dapat digulir menggunakan widget yang dibuat sesuai permintaan
3. PageView.custom() – Membuat daftar halaman yang dapat digulir dengan model anak khusus

Tidak hanya itu, dengan PageView kita dapat menambahkan transisi khusus, mengubah pengguliran halaman menjadi horizontal atau vertikal, mengatur gerakan halaman menjadi benar atau salah, dan menyesuaikan banyak hal lainnya.

d. Alat dan Bahan

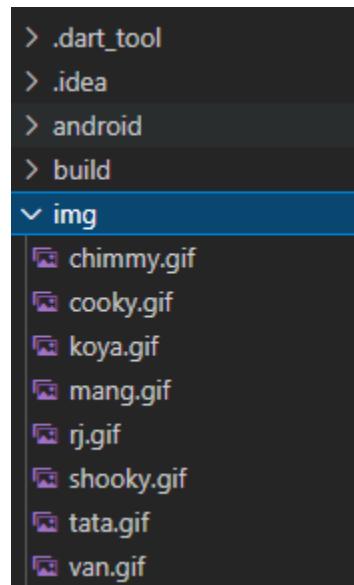
1. Kertas HVS
2. Spidol Besar

3. Bolpoin

e. Prosedur Kerja

- **Page View**

1. Download gambar pada link
2. Membuat folder img pada project, extract hasil download tadi ke dalam folder img.



3. Membuka file **pubspec.yaml**, kemudian tambahkan pada **line 48** menjadi seperti berikut:

```
47 # To add assets to your application, add an assets section, like this:  
48 assets:  
49   - img/chimmy.gif  
50   - img/cooky.gif  
51   - img/koya.gif  
52   - img/mang.gif  
53   - img/rj.gif  
54   - img/shookey.gif  
55   - img/tata.gif  
56   - img/van.gif
```

4. Tambahkan variable gambar dan warna pada class **_HomeState** seperti berikut:

```
16 final List<String> gambar = [  
17   "chimmy.gif",  
18   "cooky.gif",  
19   "koya.gif",  
20   "mang.gif",  
21   "rj.gif",  
22   "shookey.gif",  
23   "tata.gif",  
24   "van.gif"  
25 ];
```

```
27 ↴ static const Map<String, Color> colors = {  
28   'chimmy': Color(0xFF2DB569),  
29   'cooky': Color(0xFFFF386B8),  
30   'koya': Color(0xFF45CAF5),  
31   'mang': Color(0xFFB19ECB),  
32   'rj': Color(0xFFFF58E4C),  
33   'shooky': Color(0xFF46C1BE),  
34   'tata': Color(0xFFFFEA0E),  
35   'van': Color(0xFFDBE4E9),  
36 };
```

5. Tambahkan beberapa syntax pada **Widget** sehingga seperti berikut:

```
40 @override  
41 Widget build(BuildContext context) {  
42   timeDilation = 5.0;  
43   return new Scaffold(  
44     body: new Container(  
45       decoration: new BoxDecoration(  
46         gradient: new LinearGradient(  
47           begin: FractionalOffset.topCenter,  
48           end: FractionalOffset.bottomCenter,  
49           colors: [Colors.white, Colors.purple, Colors.deepPurple]), // Line  
50     ), // BoxDecoration  
51     child: new PageView.builder(  
52       controller: new PageController(viewportFraction: 0.8),  
53       itemCount: gambar.length,  
54       itemBuilder: (BuildContext context, int i) {  
55         return new Padding(  
56           padding:  
57             new EdgeInsets.symmetric(horizontal: 5.0, vertical: 50.0),  
58           child: new Material(  
59             elevation: 8.0,  
60             child: new Stack(  
61               fit: StackFit.expand,  
62               children: <Widget>[  
63                 new Hero(  
64                   tag: gambar[i],  
65                   child: new Material(  
66                     child: new InkWell(  
67
```

```

67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94

```

- Menambahkan class Halamandua seperti berikut:

```

96 class Halamandua extends StatelessWidget {
97   Halamandua({required this.gambar, required this.colors});
98   final String gambar;
99   final Color colors;
100  @override
101  Widget build(BuildContext context) {
102    return new Scaffold(
103      appBar: new AppBar(
104        title: new Text("BT21"),
105        backgroundColor: Colors.purpleAccent,
106      ), // AppBar
107      body: new Stack(
108        children: <Widget>[
109          new Container(
110            decoration: new BoxDecoration(
111              gradient: new RadialGradient(
112                center: Alignment.center,
113                colors: [Colors.purple, Colors.white, Colors.deepPurple])),

```

```

114     ), // Container
115     new Center(
116       child: new Hero(
117         tag: gambar,
118         child: new ClipOval(
119           child: new SizedBox(
120             width: 200.0,
121             height: 200.0,
122             child: new Material(
123               child: new InkWell(
124                 onTap: () => Navigator.of(context).pop(),
125                 child: new Flexible(
126                   flex: 1,
127                   child: Container(
128                     color: colors,
129                     child: new Image.asset(
130                       "img/$gambar",
131                       fit: BoxFit.cover,
132                     ), // Image.asset
133                   ), // Container
134                 ), // Flexible
135               ), // InkWell
136             ), // Material // SizedBox // ClipOval // Hero
137           ), // Center
138         ], // <Widget>[]
139       ), // Stack
140     ), // scaffold
141   }
142 }
143

```

7. Jalankan dan analisa program

f. Hasil dan Pembahasan

1. Dokumentasi tugas berupa laporan (definisi, konsep dasar gradien, pageview, dan down menu).
2. Dokumentasi tugas berupa file pdf.

g. Kesimpulan

Mahasiswa memahami mengenai definisi git, konsep dasar pengenalan dan workflow flutter.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|----|---|-------|---|---|---|
| 1 | Ketepatan waktu dan ketepatan dalam menjelaskan dari tugas ditunjang dengan bukti referensi | 1 | 2 | 3 | 4 |
| 2 | Ketepatan waktu dan ketepatan dalam menjelaskan dari tugas | 1 | 2 | 3 | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat dalam menjelaskan tugas | 1 | 2 | 3 | 4 |

| | | | | | |
|--------------------|--|------------------------------------|---|---|---|
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan dalam menjelaskan tugas | <input checked="" type="radio"/> 1 | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 23 : GRADIENT, PAGEVIEW, DROPODOWN

| | |
|----------------------------------|---|
| Pokok Bahasan | : Dropdown |
| Acara Praktikum/Pertemuan | : Minggu 6 / 3 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami konsep Gradient, Pageview, Dan Dropdown Menu pada flutter
2. Mahasiswa mampu menerapkan Gradient, Pageview, Dan Dropdown Menu pada flutter

b. Indikator

Kemampuan mahasiswa dalam memahami konsep Gradient, Pageview, Dan Dropdown Menu pada flutter

c. Dasar Teori

- **Drop Menu**

Dropdown Menu Button adalah sebuah menu yang menampilkan sebuah daftar data yang fungsinya untuk memilih suatu nilai data itu sendiri untuk di inputkan kedalam suatu data global

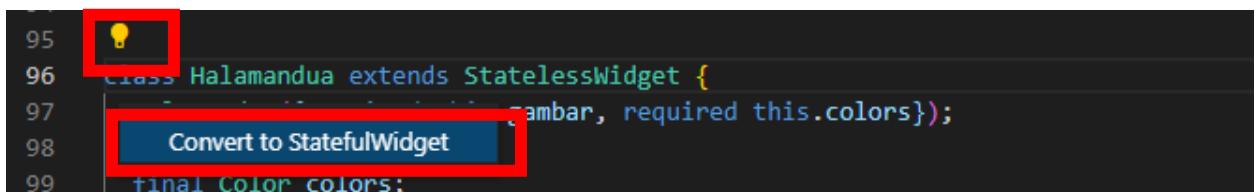
d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

- **Drop Menu**

1. Mengubah extends **StatelessWidget** menjadi **StatefulWidget** dengan cara seperti berikut:



The screenshot shows a portion of an Android Studio code editor. The code is as follows:

```
95
96 class Halamandua extends StatelessWidget {
97   final String gambar;
98   Convert to StatefulWidget
99   final Color colors;
```

A red rectangular box highlights the 'Convert to StatefulWidget' button, which is located between the closing brace of the class definition and the final color declaration. To the left of the code, there is a small icon of a lightbulb with a red border.

Hasilnya menjadi.

```
96   class Halamandua extends StatefulWidget {
97     Halamandua({required this.gambar, required this.colors});
98     final String gambar;
99     final Color colors;
100
101    @override
102    _HalamanduaState createState() => _HalamanduaState();
103  }
```

2. Menambahkan class Pilihan seperti berikut:

```
171  class Pilihan {
172    const Pilihan({required this.teks, required this.warna});
173    final String teks;
174    final Color warna;
175  }
176
```

3. Menambahkan variable listPilihan seperti berikut:

```
177  List<Pilihan> listPilihan = const <Pilihan>[
178    const Pilihan(teks: "Red", warna: Colors.red),
179    const Pilihan(teks: "Green", warna: Colors.green),
180    const Pilihan(teks: "Blue", warna: Colors.blue),
181  ]; // <Pilihan>[]
```

4. Tambahkan beberapa syntax pada class `_HalamanduaState` sehingga seperti berikut:

```
105  class _HalamanduaState extends State<Halamandua> {
106    Color warna = Colors.grey;
107
108    void _pilihannya(Pilihan pilihan) {
109      setState(() {
110        warna = pilihan.warna;
111      });
112    }
113
114    @override
115    Widget build(BuildContext context) {
116      return new Scaffold(
117        appBar: new AppBar(
118          title: new Text("BT21"),
119          backgroundColor: Colors.purpleAccent,
120          actions: <Widget>[
121            new PopupMenuButton<Pilihan>(
122              onSelected: _pilihannya,
```

```

123     itemBuilder: (BuildContext context) {
124         return listPilihan.map((Pilihan x) {
125             return new PopupMenuItem<Pilihan>(
126                 child: new Text(x.teks),
127                 value: x,
128             ); // PopupMenuItem
129         }).toList();
130     },
131 }, // PopupMenuItemButton
132 ], // <Widget>[]
133 ), // AppBar
134 body: new Stack(
135     children: <Widget>[
136         new Container(
137             decoration: new BoxDecoration(
138                 gradient: new RadialGradient(
139                     center: Alignment.center,
140                     colors: [Colors.purple, warna, Colors.deepPurple])), // Radial
141         ), // Container
142         new Center(
143             child: new Hero(
144                 tag: widget.gambar,
145                 child: new ClipOval(
146                     child: new SizedBox(
147                         width: 200.0,
148                         height: 200.0,
149                         child: new Material(
150                             child: new InkWell(
151                                 onTap: () => Navigator.of(context).pop(),
152                                 child: new Flexible(
153                                     flex: 1,
154                                     child: Container(
155                                         color: widget.colors,
156                                         child: new Image.asset(
157                                             "img/${widget.gambar}",
158                                             fit: BoxFit.cover,
159                                         ), // Image.asset
160                                         ), // Container
161                                     ), // Flexible
162                                     ), // InkWell
163                                     ))), // Material // SizedBox // ClipOval // Hero
164             ), // Center
165         ], // <Widget>[]
166     ), // Stack
167 ); // Scaffold
168 }
169 }
```

5. Jalankan dan analisa program

f. Hasil dan Pembahasan

1. Dokumentasi tugas berupa laporan (definisi, konsep dasar gradien, pageview, dan down menu).
2. Dokumentasi tugas berupa file pdf.

g. Kesimpulan

Mahasiswa memahami mengenai definisi git, konsep dasar pengenalan dan workflow flutter.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|--------------------|---|-------|---|---|---|
| | | 1 | 2 | 3 | 4 |
| 1 | Ketepatan waktu dan ketepatan dalam menjelaskan dari tugas ditunjang dengan bukti referensi | 1 | 2 | 3 | 4 |
| 2 | Ketepatan waktu dan ketepatan dalam menjelaskan dari tugas | 1 | 2 | 3 | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat dalam menjelaskan tugas | 1 | 2 | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan dalam menjelaskan tugas | 1 | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 24 : GRADIENT, PAGEVIEW, DROPDOWN

| | |
|----------------------------------|---|
| Pokok Bahasan | : Gradient, Pageview, Dropdown |
| Acara Praktikum/Pertemuan | : Minggu 6 / 4 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami konsep Gradient, Pageview, Dan Dropdown Menu pada flutter
2. Mahasiswa mampu menerapkan Gradient, Pageview, Dan Dropdown Menu pada flutter

b. Indikator

Kemampuan mahasiswa dalam memahami konsep Gradient, Pageview, Dan Dropdown Menu pada flutter

c. Dasar Teori

-

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

- a. Mahasiswa melakukan analisa code untuk setiap baris code dari penerapan gradient, pageview, dan dropdown.
- b. Menjelaskan fungsi dan arti dari setiap baris code.
- c. Analisa hasil running dari penerapan gradient, pageview, dan dropdown..
- d. Screenshot output setiap proses dari project penerapan gradient, pageview, dan dropdown.
- e. Tulisakan dalam bentuk laporan.

f. Hasil dan Pembahasan

Dokumentasi tugas berupa laporan secara individu penjelasan setiap code pada penerapan gradient, pageview, dan dropdown yang dibuktikan dengan screenshot pada proses percobaan.

g. Kesimpulan

Mahasiswa memahami penerapan gradient, pageview, dan dropdown yang dibuktikan dengan screenshot pada proses percobaan.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|--------------------|--|-------|-----|-----|-----|
| 1 | Ketepatan waktu dan ketepatan dalam menjelaskan secara detail setiap baris code pada implementasi intent implicit dan intent explicit | 1 | 2 | 3 | (4) |
| 2 | Ketepatan waktu dan ketepatan dalam menjelaskan setiap baris code pada implementasi intent implicit dan intent explicit | 1 | 2 | (3) | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat dalam menjelaskan setiap baris code pada implementasi intent implicit dan intent explicit | 1 | (2) | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan dalam menjelaskan setiap baris code pada implementasi intent implicit dan intent explicit | (1) | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 25 : FORM

| | |
|----------------------------------|---|
| Pokok Bahasan | : Form |
| Acara Praktikum/Pertemuan | : Minggu 7 / 1 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

- a. Mahasiswa mampu memahami konsep Form dalam Flutter
- b. Mahasiswa mampu menerapkan Form dalam Flutter

b. Indikator

Kemampuan mahasiswa dalam memahami dan menerapkan Form dalam Flutter.

c. Dasar Teori

Form merupakan hal yang umum ditemukan dan penting dalam sebuah aplikasi mobile. Penggunaan form juga sangat beragam, dari mulai untuk form login, register, kolom komentar, halaman order, dan banyak lagi. Karena pentingnya memahami penggunaan form pada sebuah aplikasi mobile, untuk itu dalam kesempatan ini kita akan membahas cara membuat form di Flutter menggunakan Form widget beserta komponen dan widget di dalamnya. Form widget sendiri berfungsi untuk mempermudah dalam proses pembuatan dan memberi keamanan lebih pada aplikasi flutter seperti validasi, dan aksi lainnya yang umum terdapat pada sebuah form.

Komponen Umum pada Form

Banyak sekali pilihan [flutter widget](#) yang dapat digunakan dalam sebuah form namun dalam kesempatan ini kita hanya akan membahas komponen atau widget yang umum ditemukan pada sebuah form di aplikasi flutter yaitu TextFormField, CheckBox, RadioButton, Slider Widget, dan Button.

[1] TextFormField Widget

TextFormField berfungsi sebagai input field, biasanya digunakan untuk input nama lengkap, searching, email, password dan input lainnya. Untuk contoh kode penggunaannya seperti dibawah ini

```
TextField();
```

Label dan Placeholder pada TextFormField

Untuk menambahkan label, placeholder atau icon pada TextField, kita dapat menggunakan properti **decoration**. contohnya seperti ini

```
TextField(  
    decoration: new InputDecoration(  
        hintText: "Masukan Nama Lengkap Anda",  
        labelText: "Nama Lengkap",  
        icon: Icon(Icons.people)),  
) ,
```

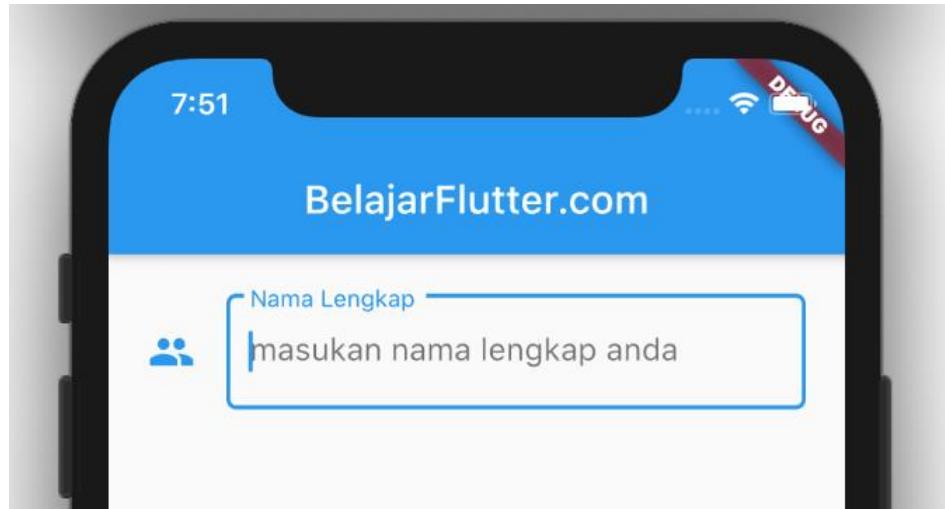
masih pada properti decoration, kita juga dapat menambah border, contohnya seperti ini

```
TextField(  
    decoration: new InputDecoration(  
        hintText: "masukan nama lengkap anda",  
        labelText: "Nama Lengkap",  
        icon: Icon(Icons.people),  
        border: OutlineInputBorder(  
            borderRadius: new BorderRadius.circular(5.0)),  
    ),  
)
```

[2] AutoFocus

Cara mengaktifkan auto fokus pada text field yaitu dengan men-set properti **autofocus** menjadi **true**

```
TextField(  
    autofocus: true,  
    ...
```



contoh TextFormField pada Flutter

[3] Password

Untuk merubah tampilan TextFormField atau TextField menjadi password, bisa dengan menggunakan properti **obscureText** dan set valuenya menjadi **true**

```
TextField(obscureText: true),
```

[4] keyboardType

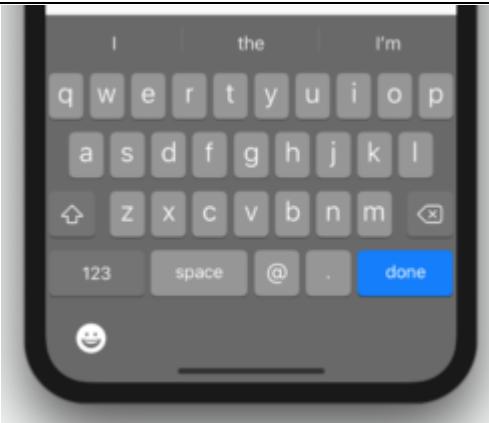
Kita juga dapat merubah tampilan tipe keyboard saat textField aktif, caranya dengan menggunakan properti **keyboardType**. Contoh

```
keyboardType: TextInputType.phone,
```

Lengkapnya seperti ini

```
TextFormField(  
  keyboardType: TextInputType.phone,  
  decoration: new InputDecoration(  
    hintText: "contoh: 0812xxxxxxxx",  
    labelText: "Nomor Telp",  
    icon: Icon(Icons.phone),  
    border: OutlineInputBorder(  
      borderRadius: new BorderRadius.circular(5.0)),  
  ),  
)
```

Pilihan dari keyboardType pada TextField dan TextFormField juga cukup beragam. Berikut opsi dan contoh pilihan tampilan keyboard pada flutter :

| Jenis | Tampilan |
|---|--|
| TextInputType.number atau TextInputType.numberWithOptions() |  |
| TextInputType.phone |  |
| TextInputType emailAddress |  |

| Jenis | Tampilan |
|------------------------|----------|
| TextInputType.url | |
| TextInputType.datetime | |

Untuk memunculkan pilihan yang tersedia silahkan tekan CTRL + space

[5] Validation

Untuk menambahkan validasi pada TextFormField cukup mudah, caranya yaitu seperti ini

```
TextField(
  decoration: new InputDecoration(
    hintText: "contoh: Susilo Bambang",
    labelText: "Nama Lengkap",
    icon: Icon(Icons.people),
    border: OutlineInputBorder(
      borderRadius: new BorderRadius.circular(5.0)),
  ),
  validator: (value) {
    if (value.isEmpty) {
      return 'Nama tidak boleh kosong';
    }
  }
),
```

```
        return null;
    },
),
```

Kode **value.isEmpty** untuk mengecek apakah nilai pada field tersebut kosong atau tidak. Kita juga dapat menambah kondisi lain sesuai kebutuhan seperti menggunakan regex untuk mengecek format email dan sejenisnya.

Properti validation akan aktif apabila fungsi validate() pada formKey di trigger

```
_formKey.currentState.validate()
```

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

1. Membuat project dengan nama minggu_7
2. Selanjutnya membuat file **form.dart** pada folder **lib**. Pertama tambahkan isyarat untuk import library material:

```
1 import 'package:flutter/material.dart';
```

3. Membuat fungsi main untuk memanggil class yang pertama kali ditampilkan:

```
3 void main() {
4   runApp(MaterialApp(
5     title: "Belajar Form Flutter",
6     home: BelajarForm(),
7   )); // MaterialApp
8 }
```

4. Kode diatas class yang pertama kali dipanggil yaitu **BelajarForm**, selanjutnya membuat class BelajarForm berupa **StatefulWidget**. Menambahkan class **BelajarForm** pada form.dart, seperti berikut :

```
10 class BelajarForm extends StatefulWidget {
11   @override
12   _BelajarFormState createState() => _BelajarFormState();
13 }
```

5. Menambahkan beberapa variable pada class **_BelajarFormState** seperti berikut :

```
15 class _BelajarFormState extends State<BelajarForm> {
16     @override
17     Widget build(BuildContext context) {
18         return Scaffold(
19             appBar: AppBar(
20                 title: Text("Belajar Form Flutter"),
21             ), // AppBar
22             body: new Container(),
23         ); // Scaffold
24     }
25 }
```

6. Tambahkan variable final `_formKey` pada class `_BelajarFormState` seperti berikut:

```
16 final _formKey = GlobalKey<FormState>();
```

7. Tambahkan beberapa syntax pada **Widget** sehingga seperti berikut:

```
18 Widget build(BuildContext context) {
19     return Scaffold(
20         appBar: AppBar(
21             title: Text("Belajar Form Flutter"),
22         ), // AppBar
23         body: Form(
24             key: _formKey,
25             child: Container(
26                 padding: EdgeInsets.all(20.0),
27                 child: Column(
28                     children: [
29                         // TextField(),
30                         TextFormField(
31                             decoration: new InputDecoration(
32                                 hintText: "contoh: Arvita Agus Kurniasari",
33                                 labelText: "Nama Lengkap",
34                                 icon: Icon(Icons.people),
35                                 border: OutlineInputBorder(
36                                     borderRadius: new BorderRadius.circular(5.0)),
37                             ), // InputDecoration
```

```

38     validator: (value) {
39       if (value!.isEmpty) {
40         return 'Nama tidak boleh kosong';
41       }
42       return null;
43     },
44   ), // TextFormField
45   RaisedButton(
46     child: Text(
47       "Submit",
48       style: TextStyle(color: Colors.white),
49     ), // Text
50     color: Colors.blue,
51     onPressed: () {
52       if (_formKey.currentState!.validate()) {}
53     },
54   ), // RaisedButton
55   ],
56   ), // Column
57   ), // Container
58   ), // Form
59   ); // Scaffold
60 }
61 }
62

```

f. Hasil dan Pembahasan

1. Pengumpulan source code melalui link commit gitlab/github pada LMS.
2. Dokumentasi tugas berupa laporan secara individu penjelasan pada prosedur kerja kemudian dikumpulkan dalam bentuk laporan PDF dan dikumpulkan di LMS

g. Kesimpulan

Mahasiswa memahami mengenai penerapan Form dalam flutter.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|----|---|-------|---|---|---|
| 1 | Ketepatan waktu dan ketepatan implementasi pada pembuatan form pada flutter | 1 | 2 | 3 | 4 |
| 2 | Ketepatan waktu dan ketepatan implementasi pada pembuatan form pada flutter | 1 | 2 | 3 | 4 |

| | | | | | |
|--------------------|--|-----|-----|---|---|
| 3 | Ketepatan waktu akan tetapi kurang tepat implementasi pada pembuatan form pada flutter | 1 | (2) | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan implementasi pada pembuatan form pada flutter | (1) | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 26 : FORM

| | |
|----------------------------------|---|
| Pokok Bahasan | : Form |
| Acara Praktikum/Pertemuan | : Minggu 7 / 2 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami konsep Form dalam Flutter
2. Mahasiswa mampu menerapkan Form dalam Flutter.

b. Indikator

Kemampuan mahasiswa dalam memahami dan menerapkan Form dalam Flutter.

c. Dasar Teori

[6] Check Box dan Switch Button

Pada flutter, untuk membuat check box bisa menggunakan Checkbox widget. Dan untuk Switch button menggunakan Switch Widget. Nilai atau value untuk kedua widget tersebut berupa boolean (true / false).

```
// contoh checkbox
Checkbox(
  value: true,
  onChanged: (value) {},
),

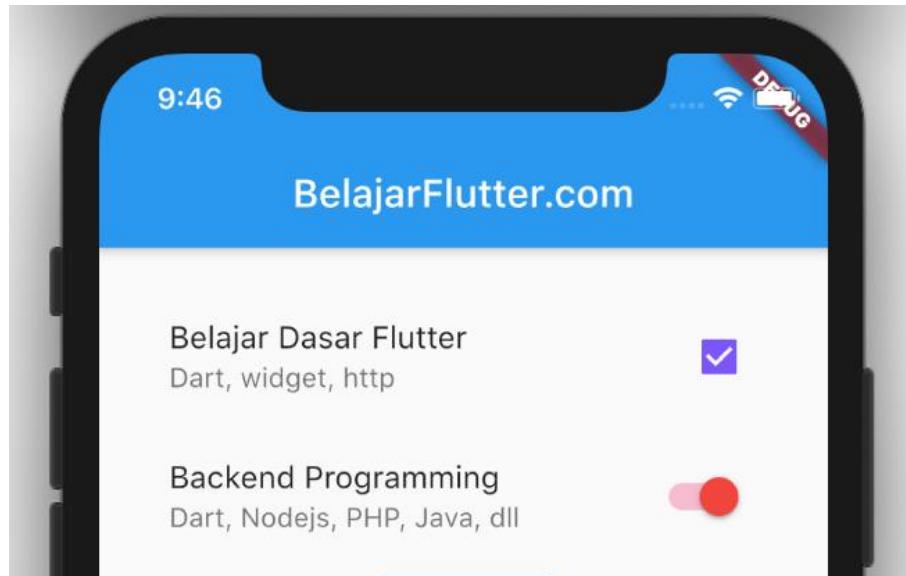
// contoh Switch
Switch(
  value: true,
  onChanged: (value) {},
),
```

Apabila membutuhkan label dan subtitle, agar lebih mudah kita dapat menggantinya menggunakan CheckboxListTile atau SwitchListTile widget

```
CheckboxListTile(
```

```
title: Text('Belajar Dasar Flutter'),  
subtitle: Text('Dart, widget, http'),  
value: true,  
activeColor: Colors.deepPurpleAccent,  
onChanged: (value) {},  
,  
SwitchListTile(  
    title: Text('Backend Programming'),  
    subtitle: Text('Dart, Nodejs, PHP, Java, dll'),  
    value: true,  
    activeTrackColor: Colors.pink[100],  
    activeColor: Colors.red,  
    onChanged: (value) {},  
,
```

properti **activeColor** untuk merubah warna widget saat value true. Agar widget ini dapat terlihat bekerja, maka nilai pada key value harus bersifat dinamis (di simpan pada state)



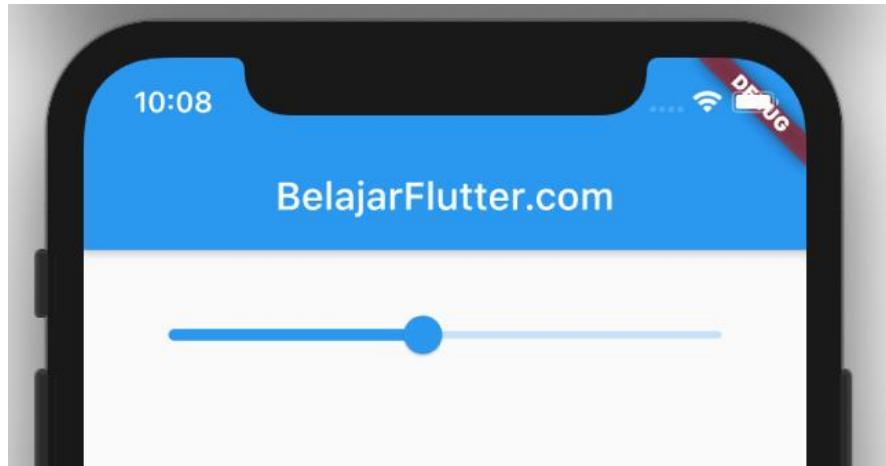
contoh tampilan checkbox dan switch button pada flutter

[7] Slider Widget

Sesuai namanya, untuk membuat input slider pada flutter silahkan gunakan Slider Widget. Nilai pada widget ini berupa Number (double). Secara sederhana penggunaanya seperti ini

```
Slider(  
    value: 25,
```

```
        min: 0,  
        max: 100,  
        onChanged: (value) {},  
,
```



contoh slider widget pada flutter

Sama seperti checkbox widget, agar dapat di-slide kita perlu merubah nilai pada properti value menjadi dinamis yaitu dengan menyimpan pada state. caranya

Buat variable baru, contoh kita beri nama **nilaiSlider**. Lalu gunakan fungsi setState pada properti onChange. Kurang lebih seperti dibawah ini

```
double nilaiSlider = 25;
```

dan pada Slider widget menjadi seperti ini

```
Slider(  
    value: nilaiSlider,  
    min: 0,  
    max: 100,  
    onChanged: (value) {  
        setState(() {  
            nilaiSlider = value;  
        });  
    },  
,
```

[8] Tombol / Button Widget

Seperti yang pernah di singgung pada pembahasan text field, penggunaan tombol biasanya untuk men-trigger validasi dan form itu sendiri. contoh kode nya seperti dibawah ini

```
RaisedButton(  
    child: Text(  
        "Submit",  
        style: TextStyle(color: Colors.white),  
    ),  
    color: Colors.blue,  
    onPressed: () {  
        if (_formKey.currentState.validate()) {}  
    },  
)
```

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

1. Buka file form.dart pada Acara 25
2. Tambahkan variable nilaiSlider, nilaiCheckBox, nilaiSwitch pada class **_BelajarFormState** seperti berikut:

```
17 |     double nilaiSlider = 1;  
18 |     bool nilaiCheckBox = false;  
19 |     bool nilaiSwitch = true;
```

3. Tambahkan beberapa syntax pada **Widget** sehingga seperti berikut:

```

62         if (value!.isEmpty) {
63             return 'Password tidak boleh kosong';
64         }
65         return null;
66     },
67     ), // TextFormField
68 ), // Padding
69 CheckboxListTile(
70     title: Text('Belajar Dasar Flutter'),
71     subtitle: Text('Dart, widget, http'),
72     value: nilaiCheckBox,
73     activeColor: Colors.deepPurpleAccent,
74     onChanged: (value) {
75         setState(() {
76             nilaiCheckBox = value!;
77         });
78     },
79 ), // CheckboxListTile
80 SwitchListTile(
81     title: Text('Backend Programming'),
82     subtitle: Text('Dart, Nodejs, PHP, Java, dll'),
83     value: nilaiSwitch,
84     activeTrackColor: Colors.pink[100],
85     activeColor: Colors.red,
86     onChanged: (value) {
87         setState(() {
88             nilaiSwitch = value;
89         });
90     },
91 ), // SwitchListTile
92 Slider(
93     value: nilaiSlider,
94     min: 0,
95     max: 100,
96     onChanged: (value) {
97         setState(() {
98             nilaiSlider = value;
99         });
100    },
101 ), // Slider
102 RaisedButton(
103     child: Text(

```

```

104     "Submit",
105         style: TextStyle(color: Colors.white),
106     ), // Text
107     color: Colors.blue,
108     onPressed: () {
109         if (_formKey.currentState!.validate()) {}
110     },
111     ), // RaisedButton
112 ],
113 ), // Column
114 ), // Container
115 ], // SingleChildScrollView
116 ), // Form
117 ); // Scaffold
118 }
119 }

```

f. Hasil dan Pembahasan

1. Pengumpulan source code melalui link commit gitlab/github pada LMS.
2. Dokumentasi tugas berupa laporan secara individu penjelasan pada prosedur kerja kemudian dikumpulkan dalam bentuk laporan PDF dan dikumpulkan di LMS

g. Kesimpulan

Mahasiswa memahami mengenai penerapan Form dalam flutter.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|--------------------|--|-------|-----|-----|-----|
| 1 | Ketepatan waktu dan ketepatan implementasi pada pembuatan form pada flutter | 1 | 2 | 3 | (4) |
| 2 | Ketepatan waktu dan ketepatan implementasi pada pembuatan form pada flutter | 1 | 2 | (3) | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat implementasi pada pembuatan form pada flutter | 1 | (2) | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan implementasi pada pembuatan form pada flutter | (1) | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 27 : FORM

| | |
|----------------------------------|---|
| Pokok Bahasan | : Form |
| Acara Praktikum/Pertemuan | : Minggu 7 / 3 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

- Mahasiswa mampu memahami konsep Form dalam Flutter
- Mahasiswa mampu menerapkan Form dalam Flutter

b. Indikator

Keberhasilan mahasiswa dalam memahami konsep Form dalam Flutter, implementasi Form pada Flutter, serta analisa source code.

c. Dasar Teori

-

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

1. Membuka file project minggu_7 sebelumnya
2. Selanjutnya buka file **main.dart** pada folder **lib**, lalu hapus semua isi kodennya. Pertama tambahkan isyarat untuk import library material:

```
1 import 'package:flutter/material.dart';
```

3. Membuat fungsi main untuk memanggil class yang pertama kali ditampilkan :

```
3 void main() {  
4   runApp(new MaterialApp(  
5     | home: new Home(),  
6     |)); // MaterialApp  
7 }
```

4. Kode diatas class yang pertama kali dipanggil yaitu **Home**, selanjutnya membuat class Home berupa **StatefulWidget**. Menambahkan class **Home** pada main.dart, seperti berikut :

```
9  < class Home extends StatefulWidget {  
10 |   // This widget is the root of your application.  
11 |   @override  
12 |   _HomeState createState() => _HomeState();  
13 | }  
14 |
```

5. Menambahkan beberapa variable pada class **_HomeState** seperti berikut :

```
15 < class _HomeState extends State<Home> {  
16 |   @override  
17 |   Widget build(BuildContext context) {  
18 |     return new Scaffold(  
19 |       appBar: new AppBar(  
20 |         leading: new Icon(Icons.list),  
21 |         title: new Text("Data diri"),  
22 |         backgroundColor: Colors.teal,  
23 |       ), // AppBar  
24 |       body: new Container(  
25 |         ), // Container  
26 |       ); // Scaffold  
27 |     }  
28 |   }  
29 | }
```

6. Tambahkan variable List <String> agama pada class **_HomeState** seperti berikut:

```
16 < List<String> agama = [  
17 |   "Islam",  
18 |   "Kristen Protestan",  
19 |   "Kristen Katolik",  
20 |   "Hindu",  
21 |   "Budha"  
22 | ];
```

7. Tambahkan variable String agama dan **_jk** pada class **_HomeState** seperti berikut:

```
24 |   String _agama = "Islam";  
25 |  
26 |   String _jk = "";
```

8. Tambahkan variable controllerNama, controllerPass, controllerMoto pada class **_HomeState** seperti berikut:

```
29 |     TextEditingController controllerNama = new TextEditingController();
30 |     TextEditingController controllerPass = new TextEditingController();
31 |     TextEditingController controllerMoto = new TextEditingController();
```

9. Tambahkan fungsi `_pilihJk` pada class `_HomeState` seperti berikut:

```
33 |     void _pilihJk(String value) {
34 |         setState(() {
35 |             _jk = value;
36 |         });
37 |     }
```

10. Tambahkan fungsi `pilihAgama` pada class `_HomeState` seperti berikut:

```
39 |     void pilihAgama(String value) {
40 |         setState(() {
41 |             _agama = value;
42 |         });
43 |     }
```

11. Tambahkan fungsi `kirimdata` pada class `_HomeState` seperti berikut:

```
45 |     void kirimdata() {
46 |         AlertDialog alertDialog = new AlertDialog(
47 |             content: new Container(
48 |                 height: 200.0,
49 |                 child: new Column(
50 |                     children: <Widget>[
51 |                         new Text("Nama Lengkap : ${controllerNama.text}"),
52 |                         new Text("Password : ${controllerPass.text}"),
53 |                         new Text("Moto Hidup : ${controllerMoto.text}"),
54 |                         new Text("Jenis Kelamin : ${_jk}"),
55 |                         new Text("Agama : ${_agama}"),
56 |                         new RaisedButton(
57 |                             child: new Text("OK"),
58 |                             onPressed: () => Navigator.pop(context),
59 |                             color: Colors.teal,
60 |                         ) // RaisedButton
61 |                     ], // <Widget>[]
62 |                 ), // Column
63 |             ), // Container
64 |         ); // AlertDialog
65 |         alertDialog.showDialog(context: context, builder: (_)
66 |             => alertDialog);
```

12. Tambahkan beberapa syntax pada **Widget** sehingga seperti berikut:

```
67  @override
68  Widget build(BuildContext context) {
69    return new Scaffold(
70      appBar: new AppBar(
71        leading: new Icon(Icons.list),
72        title: new Text("Data diri"),
73        backgroundColor: Colors.teal,
74      ), // AppBar
75      body: new ListView(
76        children: [
77          new Container(
78            padding: new EdgeInsets.all(10.0),
79            child: new Column(
80              children: <Widget>[
81                new TextField(
82                  controller: controllerNama,
83                  decoration: new InputDecoration(
84                    hintText: "Nama Lengkap",
85                    labelText: "Nama Lengkap",
86                    border: new OutlineInputBorder(
87                      borderRadius: new BorderRadius.circular(20.0))), // Out
88                ), // TextField
89                new Padding(
90                  padding: new EdgeInsets.only(top: 20.0),
91                ), // Padding
92                new TextField(
93                  controller: controllerPass,
94                  obscureText: true,
95                  decoration: new InputDecoration(
96                    hintText: "Password",
97                    labelText: "Password",
98                    border: new OutlineInputBorder(
99                      borderRadius: new BorderRadius.circular(20.0))), // Out
100               ), // TextField
101               new Padding(
102                 padding: new EdgeInsets.only(top: 20.0),
103               ), // Padding
104               new TextField(
105                 controller: controllerMoto,
106                 maxLines: 3,
107                 decoration: new InputDecoration(
108                   hintText: "Moto Hidup",
109                   labelText: "Moto Hidup",
110                   border: new OutlineInputBorder(
111                     borderRadius: new BorderRadius.circular(20.0))), // Out
112               ), // TextField
```

```
102         padding: new EdgeInsets.only(top: 20.0),
103     ), // Padding
104     new TextField(
105         controller: controllerMoto,
106         maxLines: 3,
107         decoration: new InputDecoration(
108             hintText: "Moto Hidup",
109             labelText: "Moto Hidup",
110             border: new OutlineInputBorder(
111                 borderRadius: new BorderRadius.circular(20.0))), // Out
112     ), // TextField
113     new Padding(
114         padding: new EdgeInsets.only(top: 20.0),
115     ), // Padding
116     new RadioListTile(
117         value: "laki-laki",
118         title: new Text("Laki-Laki"),
119         groupValue: _jk,
120         onChanged: (String? value) {
121             _pilihJk(value!);
122         },
123         activeColor: Colors.blue,
124         subtitle: new Text("Pilih ini jika anda Laki-laki"),
125     ), // RadioListTile
126     new RadioListTile(
127         value: "perempuan",
128         title: new Text("Perempuan"),
129         groupValue: _jk,
130         onChanged: (String? value) {
131             _pilihJk(value!);
132         },
133         activeColor: Colors.blue,
134         subtitle: new Text("Pilih ini jika anda perempuan"),
135     ), // RadioListTile
136     new Padding(
137         padding: new EdgeInsets.only(top: 20.0),
138     ), // Padding
139     new Row(
140         children: <Widget>[
141             new Text(
142                 "Agama ",
143                 style: new TextStyle(fontSize: 18.0, color: Colors.blue),
144             ), // Text
145             new Padding(
```

```

146           padding: new EdgeInsets.only(left: 15.0),
147       ), // Padding
148       DropdownButton(
149           onChanged: (String? value) {
150               pilihAgama(value!);
151           },
152           value: _agama,
153           items: agama.map((String value) {
154               return new DropdownMenuItem(
155                   value: value,
156                   child: new Text(value),
157               ); // DropdownMenuItem
158           }).toList(),
159       ) // DropdownButton
160   ], // <Widget>[]
161   ), // Row
162   new RaisedButton(
163       child: new Text("OK"),
164       color: Colors.blue,
165       onPressed: () {
166           kirimdata();
167       }) // RaisedButton

```

```

168   ], // <Widget>[]
169   ), // Column
170   ), // Container
171   ],
172   ), // ListView
173   ); // Scaffold
174 }
175 }

```

f. Hasil dan Pembahasan

1. Pengumpulan source code melalui link commit gitlab/github pada LMS.
2. Dokumentasi tugas berupa laporan secara individu penjelasan pada prosedur kerja kemudian dikumpulkan dalam bentuk laporan PDF dan dikumpulkan di LMS

g. Kesimpulan

Mahasiswa memahami mengenai penerapan Form dalam flutter.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|----|---|-------|---|---|---|
| 1 | Ketepatan waktu dan ketepatan implementasi pada pembuatan form pada flutter | 1 | 2 | 3 | 4 |
| 2 | Ketepatan waktu dan ketepatan implementasi pada pembuatan form pada flutter | 1 | 2 | 3 | 4 |

| | | | | | |
|--------------------|--|-----|-----|---|---|
| 3 | Ketepatan waktu akan tetapi kurang tepat implementasi pada pembuatan form pada flutter | 1 | (2) | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan implementasi pada pembuatan form pada flutter | (1) | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 28 : FORM

| | |
|----------------------------------|---|
| Pokok Bahasan | : Form |
| Acara Praktikum/Pertemuan | : Minggu 7 / 4 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

- Mahasiswa mampu memahami konsep Form dalam Flutter
- Mahasiswa mampu menerapkan dan meng-implementasikan konsep Form dalam Flutter

b. Indikator

Keberhasilan mahasiswa dalam memahami konsep Form dalam Flutter, implementasi konsep Form dalam Flutter, serta analisa source code.

c. Dasar Teori

-

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

1. Membuat Design menggunakan Figman untuk Form Registrasi Pendaftaran Akun
2. Mengimplementasi design Form Registrasi Pendaftaran kedalam sebuah file
form_register.dart

f. Hasil dan Pembahasan

- Dokumentasi tugas berupa laporan secara individu penjelasan pada prosedur kerja kemudian dikumpulkan dalam bentuk laporan PDF dan dikumpulkan di LMS.
- Pengumpulan source code melalui link commit gitlab/github pada LMS.

g. Kesimpulan

Mahasiswa memahami mengenai penerapan Form dalam flutter.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | | |
|--------------------|--|-------|---|---|---|--|
| 1 | Ketepatan waktu dan ketepatan implementasi pada pembuatan form pada flutter | 1 | 2 | 3 | 4 | |
| 2 | Ketepatan waktu dan ketepatan implementasi pada pembuatan form pada flutter | 1 | 2 | 3 | 4 | |
| 3 | Ketepatan waktu akan tetapi kurang tepat implementasi pada pembuatan form pada flutter | 1 | 2 | 3 | 4 | |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan implementasi pada pembuatan form pada flutter | 1 | 2 | 3 | 4 | |
| Jumlah skor | | | | | | |

ACARA 29 : STATE, ROUTE, & DEPENDENCY MANAGEMENT

| | |
|----------------------------------|---|
| Pokok Bahasan | : GetX |
| Acara Praktikum/Pertemuan | : Minggu 9 / 1 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

- 1) Mahasiswa mampu memahami konsep GetX
- 2) Mahasiswa mampu menerapkan konsep GetX

b. Indikator

Keberhasilan dalam memahami konsep GetX

c. Dasar Teori

GetX adalah solusi ekstra-ringan dan powerful untuk Flutter. Ini mengkombinasikan state management dengan performa tinggi, injeksi dependensi yang cerdas, dan route management secara singkat dan praktis.

GetX memiliki 3 prinsip dasar, yang menjadi prioritas untuk semua resource yang ada di dalamnya:

- a. **PERFORMA:** GetX fokus pada performa dan konsumsi resource minimum. GetX tidak menggunakan Stream atau ChangeNotifier.
- b. **PRODUKTIFITAS:** GetX menggunakan sintaks yang mudah dan nyaman. Tidak peduli apa yang akan dilakukan, akan selalu ada cara yang lebih mudah dengan GetX. Ini akan menghemat waktu development, dan meng-ekstrak performa maksimum pada aplikasi. Umumnya, developer akan selalu berhubungan dengan penghapusan controller dari memori. Dengan GetX, ini tidak diperlukan, karena resource akan dihapus dari memori secara default ketika tidak digunakan. Jika ingin menyimpannya kedalam memori, harus secara eksplisit mendeklarasikan "permanent: true" pada dependensi anda. Dengan begitu, selain menghemat waktu, untuk mengurangi resiko memiliki dependensi yang tidak diperlukan dalam memori. Pemuatan dependensi juga bersifat "lazy" secara default.
- c. **ORGANISASI:** GetX memungkinkan pemisahan View, Presentation Logic, Business Logic, Dependency Injection, dan Navigasi. Tidak memerlukan konteks untuk berpindah antar

halaman. Jadi, tidak lagi bergantung pada widget tree (visualisasi) untuk hal ini. Tidak perlu konteks untuk mengakses controller/bloc melalui InheritedWidget. Dengan ini, dapat memisahkan presentation logic dan business logic dari lapisan visual. Tidak perlu menginjeksi kelas Controller/Model/Bloc kedalam widget tree melalui multiprovider, untuk hal ini GetX menggunakan fitur dependency injection nya sendiri, memisahkan DI dari View secara total. Dengan GetX, harus mencari setiap fitur dalam aplikasi, memiliki kode yang bersih secara default. Ini selain untuk memfasilitasi maintenance, membuat pembagian modul, sesuatu yang hingga saat itu di Flutter tidak terpikirkan, sesuatu yang sangat mungkin. BLoC adalah permulaan awal dalam meng-organisir kode di Flutter, ini memisahkan business logic dari visualisasi. GetX adalah evolusi natural dari ini, tidak hanya memisahkan business logic, tapi juga presentation logic. Injeksi dependensi dan route juga dipisahkan sebagai bonus, dan lapisan data benar-benar terpisah secara menyeluruh. Dapat mengetahui dimana semuanya berada, dan segalanya dengan cara yang lebih mudah daripada membuat sebuah hello world. GetX adalah cara termudah, praktis, dan scalable untuk membangun aplikasi dengan performa tinggi menggunakan Flutter SDK, dengan ekosistem besar di sekelilingnya yang bekerjasama secara sempurna, mudah dipahami untuk pemula, dan akurat untuk ahli. Aman, stabil, up-to-date, dan menawarkan banyak cakupan build-in API yang tidak tersedia di dalam default Flutter SDK.

GetX tidak "bloated". Dirinya memiliki banyak fitur yang memungkinkan anda memulai programming tanpa mengkhawatirkan apapun, namun setiap fiturnya terletak didalam kontainer terpisah, dan hanya dimulai setelah digunakan. Jika anda hanya menggunakan State Management, hanya State Management yang akan di-compile. Jika hanya menggunakan routes, state management tidak akan di-compile.

GetX memiliki ekosistem yang besar, komunitas yang juga besar, banyak kolaborator, dan akan di maintenance selama Flutter ada. GetX juga mampu berjalan dengan kode yang sama di Android, iOS, Web, Mac, Linux, Windows, dan server anda. **Juga memungkinkan untuk me-reuse kode yang dibuat di frontend ke backend dengan [Get Server](#).**

Selain itu, seluruh proses development bisa di automasi secara menyeluruh, untuk keduanya (server dan frontend) menggunakan [Get CLI](#).

Tiga Pilar GetX

1. State Management
2. Route Management
3. Dependency Management

d. Alat dan Bahan

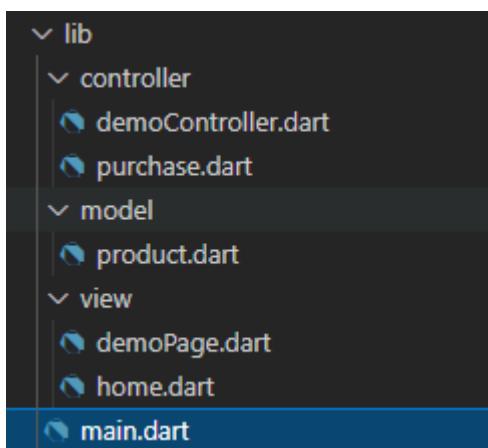
1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

- a. Membuat project dengan nama minggu_9
- b. Tambahkan library pada file pubspec.yaml seperti berikut:

```
23 dependencies:  
24   flutter:  
25     |  sdk: flutter  
26  
27  
28   # The following adds the Cupertino Icons font to your application.  
29   # Use with the CupertinoIcons class for iOS style icons.  
30   cupertino_icons: ^1.0.2  
31   get: 4.5.1  
32   get_storage: ^2.0.3
```

- c. Membuat folder dan file seperti berikut:



- d. Membuat file product.dart pada lib/model yang berisi class product.

```

1 class Product {
2     final int id;
3     final String productName;
4     final String productImage;
5     final String productDescription;
6     final double price;
7
8     Product(this.id, this.productName, this.productImage, this.productDescription,
9         this.price);
10 }

```

f. Hasil dan Pembahasan

- Dokumentasi tugas berupa laporan secara individu penjelasan pada prosedur kerja kemudian dikumpulkan dalam bentuk laporan PDF dan dikumpulkan di LMS.
- Pengumpulan source code melalui link commit gitlab/github pada LMS.

g. Kesimpulan

Mahasiswa berhasil dalam memahami konsep GetX.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|--------------------|--|-------|-----|-----|-----|
| 1 | Ketepatan waktu dan ketepatan implementasi pada konsep GetX | 1 | 2 | 3 | (4) |
| 2 | Ketepatan waktu dan ketepatan implementasi pada konsep GetX | 1 | 2 | (3) | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat implementasi pada konsep GetX | 1 | (2) | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan implementasi pada konsep GetX | (1) | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 30 : STATE, ROUTE, & DEPENDENCY MANAGEMENT

| | |
|----------------------------------|---|
| Pokok Bahasan | : State Management (GetX) |
| Acara Praktikum/Pertemuan | : Minggu 9 / 2 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami konsep State Management (GetX).
2. Mahasiswa mampu membuat code pada State Management (GetX).

b. Indikator

Keberhasilan dalam memahami konsep State Management (GetX), serta implementasi code pada State Management (GetX).

c. Dasar Teori

Get memiliki dua state manager berbeda: Simple state manager (GetBuilder) dan Reactive state manager (GetX/Obx)

- Reactive State Manager

Reactive programming bisa meng-alienasi banya orang karena katanya, sulit dimengerti. GetX mengubah reactive programming menjadi sesuatu yang cukup sederhana:

1. Tidak perlu membuat StreamController.
2. Tidak perlu membuat StreamBuilder untuk setiap variabel.
3. Tidak perlu membuat kelas untuk setiap state.
4. Tidak perlu membuat get untuk sebuah value awal (initial value).
5. Tidak perlu menggunakan generator kode.

Reactive programming dengan Get semudah menggunakan setState.

Bayangkan anda memiliki variabel nama, dan setiap kali mengubahnya, semua widget yang menggunakanannya akan berubah secara otomatis.

Ini variabel count:

```
var name = 'Jonatas Borges';
```

Untuk membuatnya "observable", anda hanya perlu menambahkan ".obs" di belakangnya:

```
var name = 'Jonatas Borges'.obs;
```

Dan didalam UI, ketika ingin menampilkan value dan update tampilan ketika value itu berubah, cukup lakukan ini:

```
Obx(() => Text("${controller.name}"));
```

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

1. Membuka project dengan nama minggu_9
2. Menambahkan file demoController.dart pada folder lib/controller.

```
1 import 'package:flutter/material.dart';
2 import 'package:get/get_state_manager/get_state_manager.dart';
3 import 'package:get_storage/get_storage.dart';
4 import '/model/product.dart';
5 import 'package:get/get.dart';
6
7 class DemoController extends GetxController {
8   var cartItems = <Product>[].obs;
9   int get cartCount => cartItems.length;
10  double get totalAmount => cartItems.fold(0.0,
11    | | (sum, element) => sum + element.price); //total fold is a dart facility
12  //add the product
13  addToCart(Product product) {
14    cartItems.add(product);
15  }
16
17  final storage = GetStorage();
18  bool get isDark => storage.read('darkmode') ?? false;
19  ThemeData get theme => isDark ? ThemeData.dark() : ThemeData.light();
20  void changeTheme(bool val) => storage.write('darkmode', val);
21 }
```

3. Menambahkan file purchase.dart pada folder lib/controller.

```
1 import 'package:get/get.dart';
2 import 'package:get/get_state_manager/get_state_manager.dart';
3 import '/model/product.dart';
4
5 class Purchase extends GetxController {
6     var products = <Product>[].obs;
7     @override
8     void onInit() {
9         fetchProducts();
10    super.onInit();
11 }
12
13 void fetchProducts() async {
14     await Future.delayed(Duration(seconds: 1));
//Call From server End
15
16     var serverResponse = [
17         Product(1, "Demo Product", "aby",
18             "This is a Product that we are going to show by getX", 300.0),
19         Product(1, "Demo Product", "aby",
20             "This is a Product that we are going to show by getX", 300.0),
21         Product(1, "Demo Product", "aby",
22             "This is a Product that we are going to show by getX", 300.0),
23         Product(1, "Demo Product", "aby",
24             "This is a Product that we are going to show by getX", 300.0),
25         Product(1, "Demo Product", "aby",
26             "This is a Product that we are going to show by getX", 300.0),
27         Product(1, "Demo Product", "aby",
28             "This is a Product that we are going to show by getX", 300.0)
29 ];
30
31     products.value = serverResponse;
32 }
33 }
```

f. Hasil dan Pembahasan

1. Dokumentasi tugas berupa laporan secara individu penjelasan pada prosedur kerja kemudian dikumpulkan dalam bentuk laporan PDF dan dikumpulkan di LMS.
2. Pengumpulan source code melalui link commit gitlab/github pada LMS

g. Kesimpulan

Mahasiswa berhasil dalam memahami konsep State Management (GetX), serta implementasi code pada State Management (GetX).

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|--------------------|--|-------|---|---|---|
| 1 | Ketepatan waktu dan ketepatan implementasi pada konsep State Management (GetX). | 1 | 2 | 3 | 4 |
| 2 | Ketepatan waktu dan ketepatan implementasi pada konsep State Management (GetX). | 1 | 2 | 3 | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat implementasi pada konsep State Management (GetX). | 1 | 2 | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan implementasi pada konsep State Management (GetX). | 1 | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 31 : STATE, ROUTE, & DEPENDENCY MANAGEMENT

| | |
|----------------------------------|---|
| Pokok Bahasan | : Dependency Management (GetX) |
| Acara Praktikum/Pertemuan | : Minggu 9 / 3 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami Dependency Management (GetX).
2. Mahasiswa mampu implementasi code Dependency Management (GetX).

b. Indikator

Keberhasilan dalam memahami Dependency Management (GetX), serta implementasi code Dependency Management (GetX).

c. Dasar Teori

Get memiliki dependency manager sederhana dan powerful yang memungkinkan mendapatkan kelas yang setara dengan Bloc atau Controller hanya dengan 1 baris kode, tanpa Provider context, tanpa inheritedWidget:

```
Controller controller = Get.put(Controller());
```

Catatan: Jika ingin menggunakan State Manager milik Get, harap untuk lebih memperhatikan [Bindings](#) api, yang mana akan membuat pengkoneksian View terhadap Controller jadi lebih mudah.

Daripada menginstansiasi kelas anda didalam kelas yang digunakan, cukup lakukan hal itu di dalam Get instance, ini akan membuatnya tersedia di semua tempat di Aplikasimu. Jadi bisa menggunakan controller (atau class Bloc) secara normal.

Tips: Dependency Management Get terpisah dari bagian lain dari package, jadi jika sebagai contoh aplikasi sudah menggunakan state manager (tidak peduli apapun itu), tidak perlu menulis ulang sama sekali, bisa menggunakan dependency injection tanpa masalah.

```
controller.fetchApi();
```

Bayangkan anda bermigrasi melewati route yang sangat banyak, dan membutuhkan data yang tertinggal didalam controller jauh di belakang route sebelumnya, akan butuh state manager

dikombinasikan dengan Provider atau Get_it, benar kan? Tidak dengan Get. hanya perlu meminta Get untuk "menemukan" controllernya, anda tidak perlu dependensi tambahan:

```
Controller controller = Get.find();  
// Ya, terlihat seperti Sulap, Get akan menemukan controller anda, dan akan  
mengantarkannya ke lokasi anda.  
// Anda bisa memiliki 1 juta controller terinisialisasi, Get akan selalu memberimu  
controller yang tepat.
```

Dan setelahnya anda bisa memperoleh data yang tertinggal sebelumnya:

```
Text(controller.textFromApi);
```

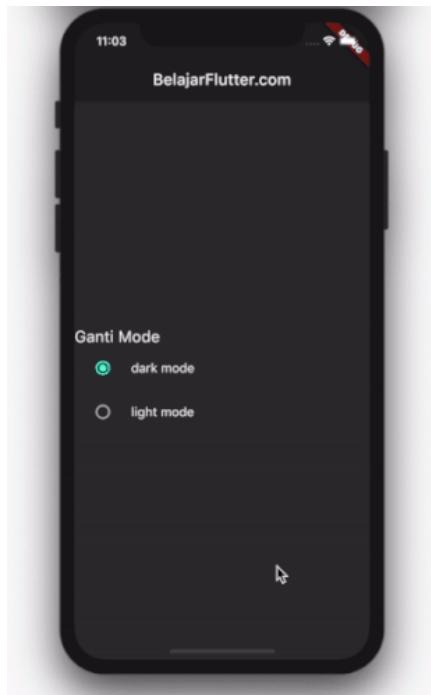
Ganti Tema hanya dengan 1 Baris kode

Membuat dynamic theme juga menjadi mudah dengan GetX, hanya dengan satu baris kode menggunakan `Get.changeTheme` dan selesai

```
Get.changeTheme(ThemeData.light());
```

atau bisa juga dengan

```
Get.changeThemeMode(ThemeMode.dark);
```



Membuat Multi Bahasa dan Locales

Kita juga dapat membuat aplikasi multi bahasa dengan menggunakan class translations seperti dibawah ini

```
import 'package:get/get.dart';

class Messages extends Translations {
    @override
    Map<String, Map<String, String>> get keys => {
        'en_US': {
            'hello': 'Hello World',
        },
        'id_ID': {
            'hello': 'Hallo Dunia',
        }
    };
}

Set Local
return GetMaterialApp(
    translations: Messages(), // your translations
    locale: Locale('en', 'US'), // translations will be displayed in that locale
    fallbackLocale: Locale('id', 'ID'), // set default jika locale yang dipilih tidak valid.
);

menampilkan text

Text('hello'.tr);
```

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

1. Membuka project dengan nama minggu_9
2. Menambahkan file demoPage.dart pada folder lib/view.

```

1 import 'package:flutter/material.dart';
2 import 'package:get/get.dart';
3 import '/controller/demoController.dart';
4
5 class DemoPage extends StatelessWidget {
6   final DemoController ctrl = Get
7     .find(); //getting the cart controller , you can show amount or anything
8   @override
9   Widget build(BuildContext context) {
10    return Scaffold(
11      appBar: AppBar(
12        title: Text('Demo Page'),
13      ), // AppBar
14      body: Center(
15        child: Column(
16          mainAxisAlignment: MainAxisAlignment.center,
17          crossAxisAlignment: CrossAxisAlignment.end,
18          children: [
19            Padding(
20              padding: const EdgeInsets.all(8.0),
21              child: Text(Get.arguments
22                SwitchListTile(
23                  value: ctrl.isDark,
24                  title: Text("Touch to change ThemeMode"),
25                  onChanged: ctrl.changeTheme,
26                ), // SwitchListTile
27                ElevatedButton(
28                  onPressed: () => Get.snackbar(
29                    "Snackbar", "Hello this is the Snackbar message",
30                    snackPosition: SnackPosition.BOTTOM,
31                    colorText: Colors.white,
32                    backgroundColor: Colors.black87),
33                    child: Text('Snack Bar')), // ElevatedButton
34                    ElevatedButton(
35                      onPressed: () => Get.defaultDialog(
36                        title: "Dialogue",
37                        content: Text(
38                          'Hey, From Dialogue',
39                        )), // Text
40                        child: Text('Dialogue')), // ElevatedButton
41                        ElevatedButton(
42                          onPressed: () => Get.bottomSheet(Container(
43                            height: 150,
44                            color: Colors.white,
45                            child: Text(

```

```

48             'Hello From Bottom Sheet',
49             style: TextStyle(fontSize: 30.0),
50         ), // Text
51     )), // Container
52     child: Text('Bottom Sheet')), // ElevatedButton
53     ElevatedButton(
54         //off named because we want to remove the page stack
55         onPressed: () =>
56             Get.offNamed('/'), //for app remove all stack use Get.offAll
57         child: Text('Back To Home')), // ElevatedButton
58     ],
59     ), // Column
60   ), // Center
61 ); // Scaffold
62 }
63 }

```

3. Menambahkan file home.dart pada folder lib/view.

```

1  import 'package:flutter/material.dart';
2  import 'package:get/get_state_manager/get_state_manager.dart';
3  import 'package:get/instance_manager.dart';
4  import '/controller/demoController.dart';
5  import '/controller/purchase.dart';
6  import 'package:get/get_navigation/get_navigation.dart';
7  import 'package:get/get.dart';
8  import 'package:get/get_core/get_core.dart';
9
10 class HomePage extends StatelessWidget {
11     final purchase = Get.put(Purchase()); // controller inti
12     DemoController cart = Get.find(); // controller cart
13
14     @override
15     Widget build(BuildContext context) {
16         return Scaffold(
17             appBar: AppBar(
18                 title: Text('Home'),
19             ), // AppBar
20             bottomSheet: SafeArea(
21                 child: Card(
22                     elevation: 12.0,
23                     margin: EdgeInsets.zero,
24                     child: Container(
25                         decoration: BoxDecoration(
26                             color: Colors.blue,

```

```

27   ),
28   height: 65,
29   child: Padding(
30     padding: const EdgeInsets.all(8.0),
31     child: Row(
32       mainAxisAlignment: MainAxisAlignment.spaceBetween,
33       children: [
34         Stack(
35           children: [
36             Icon(Icons.shopping_cart_rounded,
37               size: 40, color: Colors.white), // Icon
38             Positioned(
39               right: 5,
40               child: Container(
41                 height: 20,
42                 width: 20,
43                 decoration: BoxDecoration(
44                   shape: BoxShape.circle,
45                   color: Colors.red,
46                 ), // BoxDecoration
47                 child: Center(
48                   child: GetX<DemoController>(builder: (controller) {
49                     return Text(
50                       '${controller.cartCount}',
51                       style: TextStyle(
52                         color: Colors.white,
53                         fontWeight: FontWeight.bold,
54                         fontSize: 11.0), // TextStyle
55                     ); // Text
56                   }, // GetX
57                 ), // Center
58                 ), // Container
59               ), // Positioned
60             ],
61           ), // Stack
62           GetX<DemoController>(builder: (controller) {
63             return Text(
64               'Total Amount - ${controller.totalAmount}',
65               style: TextStyle(
66                 color: Colors.white,
67                 fontWeight: FontWeight.w300,
68                 fontSize: 16.0), // TextStyle
69               ); // Text
70             }, // GetX

```

```

71           IconButton(
72             onPressed: () => Get.toNamed('/cart',
73               arguments:
74                 "Home Page To Demo Page -> Passing arguments"),
75               icon: Icon(
76                 Icons.arrow_forward_ios_rounded,
77                 color: Colors.white,
78               )) // Icon // IconButton
79           ],
80         ), // Row
81       ), // Padding
82     ), // Container
83   ), // Card
84 ), // SafeArea
85 body: Container(
86   child: GetX<Purchase>(builder: (controller) {
87     return ListView.builder(
88       itemCount: controller.products.length,
89       itemBuilder: (context, index) => Padding(
90         padding: const EdgeInsets.all(8.0),
91         child: Card(
92           elevation: 2.0,
93           shape: RoundedRectangleBorder(
94             borderRadius: BorderRadius.circular(6.0),
95           ), // RoundedRectangleBorder
96           child: Column(
97             children: [
98               Image.network(
99                 'https://img.alicdn.com/tfs/TB1e.XyReL2gK0jSZFmXXc7iXXa-990-400.png',
100                 fit: BoxFit.cover,
101               ), // Image.network
102             Row(
103               mainAxisAlignment: MainAxisAlignment.spaceBetween,
104               crossAxisAlignment: CrossAxisAlignment.center,
105               children: [
106                 Expanded(
107                   child: Padding(
108                     padding: const EdgeInsets.all(8.0),
109                     child: Column(
110                       mainAxisAlignment: MainAxisAlignment.start,
111                       crossAxisAlignment:
112                         CrossAxisAlignment.start,
113                       children: [
114                         Text(
115                           controller.products[index].productName,

```

```

116         style: TextStyle(
117             color: Colors.blue,
118             fontWeight: FontWeight.bold,
119             fontSize: 16.0), // TextStyle
120     ), // Text
121     Text(
122         controller.products[index]
123             .productDescription,
124             style: TextStyle(
125                 color: Colors.grey[600],
126                 fontWeight: FontWeight.normal,
127                 fontSize: 12.0)), // TextStyle //
128     ],
129     ), // Column
130     ), // Padding
131     ), // Expanded
132     Padding(
133         padding: const EdgeInsets.all(8.0),
134         child: ElevatedButton(
135             onPressed: () => cart
136                 .addToCart(controller.products[index]),
137             child: Text(
138                 'Shop Now',
139                 style: TextStyle(
140                     fontWeight: FontWeight.normal,
141                     fontSize: 12.0), // TextStyle
142             ), // Text // ElevatedButton
143             ), // Padding
144             ],
145             ), // Row
146             ],
147             ), // Column
148             ), // Card
149             )); // Padding // ListView.builder
150         }, // GetX
151         ), // Container
152     ); // Scaffold
153 }
154 }
```

f. Hasil dan Pembahasan

- Dokumentasi tugas berupa laporan secara individu penjelasan pada prosedur kerja kemudian dikumpulkan dalam bentuk laporan PDF dan dikumpulkan di LMS.
- Pengumpulan source code melalui link commit gitlab/github pada LMS.

g. Kesimpulan

Mahasiswa berhasil dalam memahami konsep Dependency Management (GetX), serta implementasi code Dependency Management (GetX).

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|--------------------|---|-------|---|---|---|
| 1 | Ketepatan waktu dan ketepatan implementasi pada konsep Dependency Management (GetX). | 1 | 2 | 3 | 4 |
| 2 | Ketepatan waktu dan ketepatan implementasi pada konsep Dependency Management (GetX). | 1 | 2 | 3 | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat implementasi pada konsep Dependency Management (GetX). | 1 | 2 | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan implementasi pada konsep Dependency Management (GetX). | 1 | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 32 : STATE, ROUTE, & DEPENDENCY MANAGEMENT

| | |
|----------------------------------|---|
| Pokok Bahasan | : Route Management (GetX) |
| Acara Praktikum/Pertemuan | : Minggu 9 / 4 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami Route Management (GetX)
2. Mahasiswa mampu implementasi code Route Management (GetX)

b. Indikator

Keberhasilan dalam memahami Route Management (GetX), serta implementasi code Route Management (GetX).

c. Dasar Teori

Jika ingin menggunakan routes/snackbars/dialogs/bottomsheets tanpa context, GetX luar biasa cocok untuk, lihat ini:

Tambahkan "Get" sebelum MaterialApp, mengubahnya menjadi GetMaterialApp

```
GetMaterialApp( // Sebelumnya: MaterialApp(  
    home: MyHome(),  
)
```

Pindah ke halaman baru:

```
Get.to(NextScreen());
```

Pindah ke halaman baru menggunakan nama. Baca detail lebih lanjut tentang penamaan route [disini](#)

```
Get.toNamed('/details');
```

Untuk menutup snackbar, dialog, bottomsheet, atau apapun yang normalnya tutup menggunakan Navigator.pop(context);

```
Get.back();
```

Untuk pergi ke halaman baru dan mencegah user kembali ke halaman sebelumnya (biasanya digunakan untuk SplashScreen, LoginScreen, dsb).

```
Get.off(NextScreen());
```

Untuk pergi ke halaman baru dan batalkan navigasi sebelumnya (berguna untuk shopping cart, polls, dan test).

```
Get.offAll(NextScreen());
```

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

1. Membuka project dengan nama minggu_9
2. Selanjutnya buka file **main.dart** pada folder **lib**, lalu hapus semua isi kodennya. Pertama tambahkan isyarat untuk import library material:

```
1 import 'package:flutter/material.dart';
2 import 'package:get/get.dart';
3 import 'package:get_storage/get_storage.dart';
4 import 'controller/demoController.dart';
5 import 'view/demoPage.dart';
6 import 'view/home.dart';
```

3. Membuat fungsi main untuk memanggil class yang pertama kali ditampilkan :

```
8 void main() async {
9   await GetStorage.init();
10  runApp(MyApp());
11 }
```

4. Kode diatas class yang pertama kali dipanggil yaitu **MyApp**, selanjutnya membuat class Home berupa **StatelessWidget**. Menambahkan class **MyApp** pada main.dart, seperti berikut:

```
13 class MyApp extends StatelessWidget {
14   // This widget is the root of your application.
15   final DemoController ctrl = Get.put(DemoController());
```

```

16  @override
17  Widget build(BuildContext context) {
18    return SimpleBuilder(builder: (_) {
19      // for darkmode instant change
20      return GetMaterialApp(
21        //for navigation dont forget to use GetMaterialApp
22        title: 'GetX',
23        theme: ctrl.theme,
24        debugShowCheckedModeBanner: false,
25        initialRoute: '/',
26        routes: {
27          //routes for named navigation
28          '/': (context) => HomePage(),
29          '/cart': (context) => DemoPage(),
30        },
31      ); // GetMaterialApp
32    }); // SimpleBuilder
33  }
34 }

```

f. Hasil dan Pembahasan

1. Dokumentasi tugas berupa laporan secara individu penjelasan pada prosedur kerja kemudian dikumpulkan dalam bentuk laporan PDF dan dikumpulkan di LMS.
2. Pengumpulan source code melalui link commit gitlab/github pada LMS

g. Kesimpulan

Mahasiswa memahami konsep dasar Route Management (GetX) dengan mampu memahami setiap baris code dari implementasi Route Management (GetX).

h. Rubrik Penilaian

| No | Indikator | Skor* | | | | |
|--------------------|---|-------|---|---|---|--|
| 1 | Ketepatan waktu dan ketepatan implementasi pada konsep Route Management (GetX) | 1 | 2 | 3 | 4 | |
| 2 | Ketepatan waktu dan ketepatan implementasi pada konsep Route Management (GetX) | 1 | 2 | 3 | 4 | |
| 3 | Ketepatan waktu akan tetapi kurang tepat implementasi pada konsep Route Management (GetX) | 1 | 2 | 3 | 4 | |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan implementasi pada konsep Dep Route Management (GetX) | 1 | 2 | 3 | 4 | |
| Jumlah skor | | | | | | |

ACARA 33 : REST API

| | |
|----------------------------------|---|
| Pokok Bahasan | : Rest API |
| Acara Praktikum/Pertemuan | : Minggu 10 / 1 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami Rest API
2. Mahasiswa mampu implementasi Rest API

b. Indikator

Keberhasilan dalam memahami Rest API, serta implementasi code Rest API.

c. Dasar Teori

HTTP, REST API, JSON

Pada umumnya project yang dibuat dengan Flutter akan berkomunikasi dengan API pada sebuah server, dan kebanyakan API ini dibuat dengan desain REST dan data akan ditampilkan dalam format JSON.

Komunikasi Asynchronous

Ketika aplikasi yang kita buat berkomunikasi ke / dari remote server menggunakan HTTP, hal tersebut dilakukan secara asynchronous, aplikasi tersebut tidak akan berhenti setelah mengirim request ke server tersebut, Bahasa Dart mendukung penuh pemrograman asynchronous, termasuk Futures, Paket flutter yang menggunakan Future memungkinkan developer untuk melakukan komunikasi melalui HTTP secara asynchronous, setiap kita berkomunikasi dengan server menggunakan HTTP, kita tidak akan berhenti melakukan sesatu di aplikasi tersebut, tetapi kita akan memproses success atau error dari response yang kita terima.

HTTP

Hypertext Transfer Protocol dirancang untuk memungkinkan client dan server saling berkomunikasi. HTTP bekerja dengan protokol request - response antara client dan server, Protokol

menggambarkan bagaimana mesin berkomunikasi satu dengan lainnya menggunakan pesan, dan protokol tersebut mendefenisikan format dari pesan yang digunakan.

Request

Request adalah permintaan yang kita kirim ke server. Perhatikan contoh request berikut:



Response

Response adalah balasan yang diberikan server yang sesuai dengan permintaan kita.



Methods

Metode - metode yang terdapat pada HTTP adalah POST, GET, PUT, PATCH, dan DELETE. Metode GET yang paling sering digunakan, karena kita akan lebih sering mengakses data dari pada merubah data yang ada pada server. Metode GET digunakan untuk melakukan request data dari server sedangkan metode POST digunakan untuk mengirim data ke server, untuk mengupdate atau

menyimpan data, metode PUT digunakan untuk melakukan update pada server, dan metode DELETE digunakan untuk melakukan penghapusan data pada server.

URI

Ini adalah alamat dari request dengan path dan server yang spesifik.

Contohnya:

<https://www.cnn.com/2019/03/16/politics/beto-orourke-campaigning-while-driving/index.html>

Query Parameters.

HTTP mengizinkan kita untuk memberikan/melewatkkan informasi ke server pada URL menggunakan Query Parameters.

Contohnya:

<http://localhost:4200/sockjs-node/info?t=1498649243238>

Matrix Parameter

HTTP mengizinkan kita untuk memberikan/melewatkkan informasi ke server pada URL menggunakan matrix parameter,

Contohnya:

<http://localhost:4200/sockjs-node/info;t=1498649243238>

Matrix parameter mirip dengan query string tetapi menggunakan pattern yang berbeda. Matrix parameter juga berbeda karena tidak memiliki ‘?’ dan matrix parameter dapat di cache.

Path Parameter.

HTTP mengizinkan kita untuk memberikan/melewatkkan informasi ke server pada URL menggunakan Path parameters.

Contohnya:

<http://localhost:4200/api/badges/9243238>

URI Encoding

Ada beberapa karakter yang tidak dapat disertakan pada URL (sebagai contoh adalah spasi) dan beberapa karakter yang mempunyai arti khusus pada URL. Untuk mengatasi hal ini, sintak pada URL pada parameter akan di encoding untuk memastikan bahwa URL yang akan di request adalah valid.

Contohnya: karakter 'spasi' antara Atlantic dan City di encoded menjadi %20

[https://trailapi-trailapi.p.mashape.com/?q\[city_cont\]=Atlantic%20City](https://trailapi-trailapi.p.mashape.com/?q[city_cont]=Atlantic%20City)

Status

| Code | | Name | Deskripsi |
|-------------|-----|----------------------|--|
| 1xx | | Informational | |
| 2xx | | Success | |
| | 200 | OK | |
| 3xx | | Redirect | |
| | 301 | Moved Permanently | |
| | 302 | Moved Temporarily | |
| 4xx | | Request Error | |
| | 400 | Bad request | Request tidak dapat dimengerti oleh server. |
| | 403 | Forbidden | User tidak memiliki hak untuk melakukan operasi request. |

| | | | |
|-----|-----|-----------------------|--|
| | 404 | Not Found | Resource yang di request pada URi tidak dapat ditemukan. |
| | 405 | Method not allowed | Metode request tidak diizinkan pada resource tertentu. |
| 5xx | | Server error | |
| | 500 | Internal server error | Terjadi kesalahan ada server, request tidak dapat dilakukan. |
| | 503 | Service unavailable | Server tidak tersedia, biasanya karena overload atau maintenence |

Status adalah bagian dari response, status menandakan apakah request berhasil di proses atau tidak.

Berikut adalah beberapa status kode dari HTTP:

Header

HTTP Header mengizinkan client dan server untuk melewatkkan beberapa informasi dengan request atau response. Header pada request terdiri dari pasangan key dan value, key bersifat case insensitive.

Contoh:

```
Host: localhost:8000
Connection: keep-alive
Content-Type: application/json
```

Body

HTTP body mengizinkan client dan server untuk melewatkkan informasi tambahan dengan request atau response setelah header.

body pada request tidak selalu diminta karena informasi pada body tidak selalu dibutuhkan. Request HTTP dengan metode GET, DELETE biasanya tidak memerlukan body, sedangkan pada request HTTP dengan metode POST, PUT dan PATCH memerlukan informasi ini.

body pada response digunakan untuk mengembalikan informasi dan itu bisa sangat besar, dengan jumlah data yang besar.

API

Ketika seseorang membuat API nya tersedia ke internet, mereka menulis code untuk API tersebut dan mempublishnya ke server HTTP mereka. APIs juga dikenal sebagai web services.

Kebanyakan API menggunakan arsitektur REST, yang mana itu adalah bentuk bagaimana kamu akan berkomunikasi dengan server melalui HTTP. API yang sesuai dengan arsitektur REST kebanyakan bekerja dengan cara yang sama, dengan alamat web (URIs) dan metode HTTP yang sama, kemiripan ini sangat membantu ketika kita akan bekerja dari satu API ke lainnya.

REST

REST merupakan singkatan dari Representational State Transfer. REST memberikan kita panduan desain tingkat tinggi dan kita hanya akan memikirkan bagaimana kita mengimplementasikannya.

REST APIs harus stateless.

Dahulu kala, aplikasi web digunakan untuk menyimpan data session untuk user. Sebagai contoh, user akan melakukan login dan hal ini akan memulai session dan informasi bisa disimpan pada session ini sampai user melakukan logout, data pada session ini bisa berupa informasi user tersebut, hak akses apa yang dia punya dan informasi lain yang diperlukan.

Sekarang, dengan API dan REST yang lebih modern, akses ke server dikontrol melalui token atau api keys. Juga, setiap pemanggilan API bersifat stateless - setiap request dari client ke server bersifat mandiri dan berisi semua data yang diperlukan untuk mengidentifikasi siapa yang membuat request dan semua data request itu sendiri untuk melakukan operasi.

Menentukan User - Siapa yang membuat request ke API ?

Tokens

Pada aplikasi dengan halaman login, ketika user login terjadi, user diberikan token sementara untuk akses, Token ini dienkripsi dan berisi informasi tentang user dan token tersebut (misalnya kapan token akan berakhir/expires). Token ini dapat diperbarui (refresh) setiap waktu yang ditentukan (misalnya 10 menit). Kapan pun API dipanggil dari beberapa perangkat ke server, token tersebut harus disertakan pada request header setiap request ke server yang terjadi, jika token tidak tersedia atau tidak sah (invalid) maka server akan mengembalikan error code (biasanya kode HTTP 401 atau 403). Jika token valid maka server tahu bahwa user yang sah sedang menggunakan aplikasi tersebut, server memiliki info tentang user dari token dan API dapat melakukan operasi yang diminta.

API Keys

Jika user tidak perlu melakukan login setiap kali aplikasi digunakan, API key memungkinkan user yang terdaftar (misalnya CAT API user) untuk diidentifikasi pada header HTTP sebagai user yang sah pada setiap request yang terjadi ke server. Sama seperti Token, ini juga akan dilakukan validasi dan server akan mengembalikan kode error jika terjadi masalah.

No User Identification.

Terkadang orang mem-publish API dimana untuk mengaksesnya tidak diperlukan informasi dari user yang mengakses.

Bagaimana REST menggunakan URLs

Pada REST, URL digunakan untuk menentukan resource apa yang akan digunakan.

Base URL

Base URL adalah bagian pertama dari API, tanpa bagian dari REST. Bagian dari REST ada setelah base URL. Base URL biasanya sebagai berikut:

1. Domain, contohnya: www.google.com, www.facebook.com, dll
2. Optional sufix ‘api’ untuk mengindikasikan bahwa path ini hanya untuk API
3. Optional sufix untuk nama dari aplikasi, API yang ditulis untuk aplikasi tersebut.
4. Optional versi dari API

Sebagai contoh:

[www://dummy.restapiexample.com/api/v1](http://dummy.restapiexample.com/api/v1)

URL & Paths

URL dari REST API bisa disusun dari beberapa bagian path, contohnya:
<http://www.example.com/customers/1234/orders/821/lineitem>

Artinya:

1. Pergi ke customer 1234
2. kemudian order 821 untuk customer tersebut
3. kemudian pergi ke line item 1 untuk order tersebut.

Bagaimana REST meenggunakan metode HTTP

Pada REST, metode HTTP digunakan untuk menggambarkan apa yang kita kerjakan. Mengambil data, post data baru (menciptakannya), mengirim data (update) dan menghapus data.

Mengakses data dengan REST API

- URI
 - Tentukan data yang akan kita akses.
 - daftar item-item.
 - base url + nama resource
Contohnya: www.example.com/products, biasanya ini akan mengembalikan berbagai jenis product.
 - daftar item bisa jadi milik entity lainnya,
Contohnya:
<http://www.example.com/customers/1234/orders/>, Akan mengembalikan daftar order untuk customer 1234
<http://www.example.com/customers/1234/orders/123/lineItems>, Akan mengembalikan line items untuk order 123 dari customer 1234
 - Daftar item yang dicari
 - URLnya mirip dengan cotoh yang sebelumnya, ditambah dengan informasi tambahan pada akhir untuk menentuan pencarian.

- informasi tambahan
- kita dapat menambahkan query string atau matrix / path parameter pada akhir url.
Contohnya:
<http://www.example.com/products?name=mark>
- kita bisa menambahkan '/search' dan kriteria pencarian pada akhir URL. Contohnya:
<http://www.example.com/products/search/name/mark>, yang akan mencari products dengan nama mark
- Item tunggal
 - URL mirip dengan contoh sebelumnya ditambah slash kemudian identifier untuk mengidentifikasi item tersebut. Contohnya:
<http://www.example.com/products/66543>, akan mengembalikan product 66543
 - Item tunggal juga bisa menjadi milik entity lainnya, contohnya:
<http://www.example.com/customers/3324/orders/8769>, yang akan mengembalikan item tunggal order 8769 untuk customer 3324
- HTTP Method
kita harus menggunakan metode HTTP GET untuk mengakses data melalui REST API
- HTTP Body
tidak digunakan

Insert DATA dengan REST API

- URI
 - Tentukan data apa yang akan kita masukkan.
 - ini akan sama dengan URL sebelumnya. Contohnya:
 - <http://www.example.com/products>
 - <http://www.example.com/customers/33245/orders/>
 - Atau <http://www.example.com/ordres> (implementasi terbuka)
- HTTP Method
Kita harus menggunakan metode HTTP GET untuk memasukkan (menciptakan) data melalui REST API
- HTTP Body
umumnya kita meletakkan data yang dierlukan untuk dimasukkan kedalam request body.

Update Data dengan REST API

- URI
 - Identifikasi data apa yang kita update
 - ini akan sama dengan URL sebelumnya untuk mengakses item tunggal. Contohnya:
 - o <http://www.example.com/products/66432>
 - o <http://www.example.com/customers/33245/orders/8769>
- HTTP Method
 - kita menggunakan metode HTTP PUT untuk mengupdate data melalui REST API
- HTTP body
 - umumnya kita meletakkan data yang dierlukan untuk dimasukkan kedalam request body.

Menghapus Data dengan REST API

- URI
 - Identifikasi data apa yang akan kita hapus
 - ini akan sama dengan URL sebelumnya untuk mengakses item tunggal. Contohnya:
 - o <http://www.example.com/products/66432>
 - o <http://www.example.com/customers/33245/orders/8769>
- HTTP Method
 - kita harus menggunakan metode HTTP DELETE untuk menghapus data melalui REST API
- HTTP Body
 - tidak digunakan.

JSON

JSON merupakan singkatan dari JavaScript Object Notation. JSON adalah format yang digunakan untuk melewatkkan data antara client dan server (kedua arah, sever ke client, client ke server),JSON juga format data yang sama yang digunakan oleh javascript, menggunakan coma ',' untuk memisahkan item-itemnya dan colon ":" untuk memisahkan property nama dan property datanya, JSON menggunakan tanda kurung yang berbeda untuk menentukan object dan array.

JSON untuk melewatkkan data sebagai object.

Tandan kurung '{' dan '}' digunakan untuk menunjukkan awal dan akhir dari suatu objek.

```
{“name”:”John”, “age”: 31, “City”: “Indonesia”}
```

JSON untuk melewaskan array

Tanda kurung ‘[‘ dan ‘]’ digunakan untuk menunjukkan awal dan akhir dari array.

```
[“Ford”, “BMW”, “Fiat”]
```

JSON untuk melewaskan array dari beberapa objek.

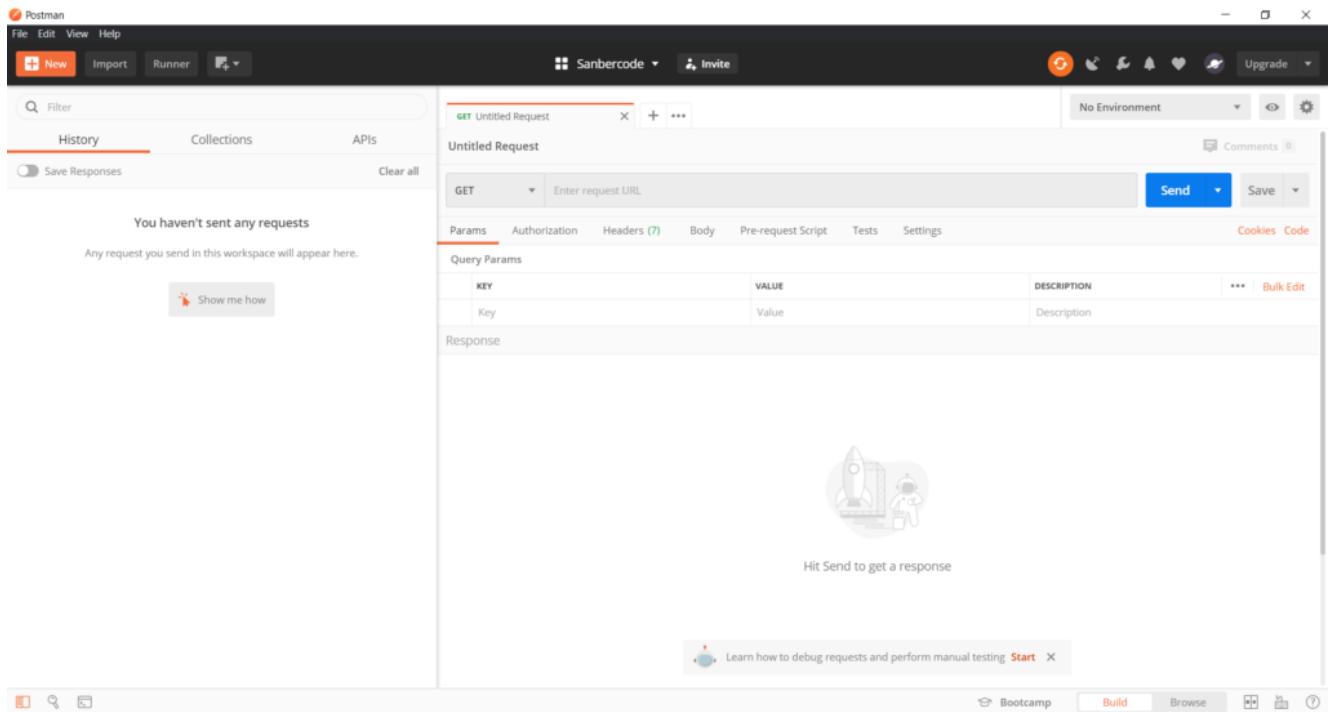
Tanda kurung dikombinasikan untuk menciptakan object car, yang mempunyai dua properti ‘Nissan’ dan ‘Ford’, dimana setiap properti memiliki model-model yang tersusun dalam array.

```
{
  “cars”: [
    {"model : “Sentra”, “doors”:4}, {"model”: “Maxima”, “doors”:4}
  ],
  “Ford”: [
    {"model”: “Taurus”, “doors”: 4}, {"model” : “Escort”, “doors”:4}
  ]
}
```

Postman

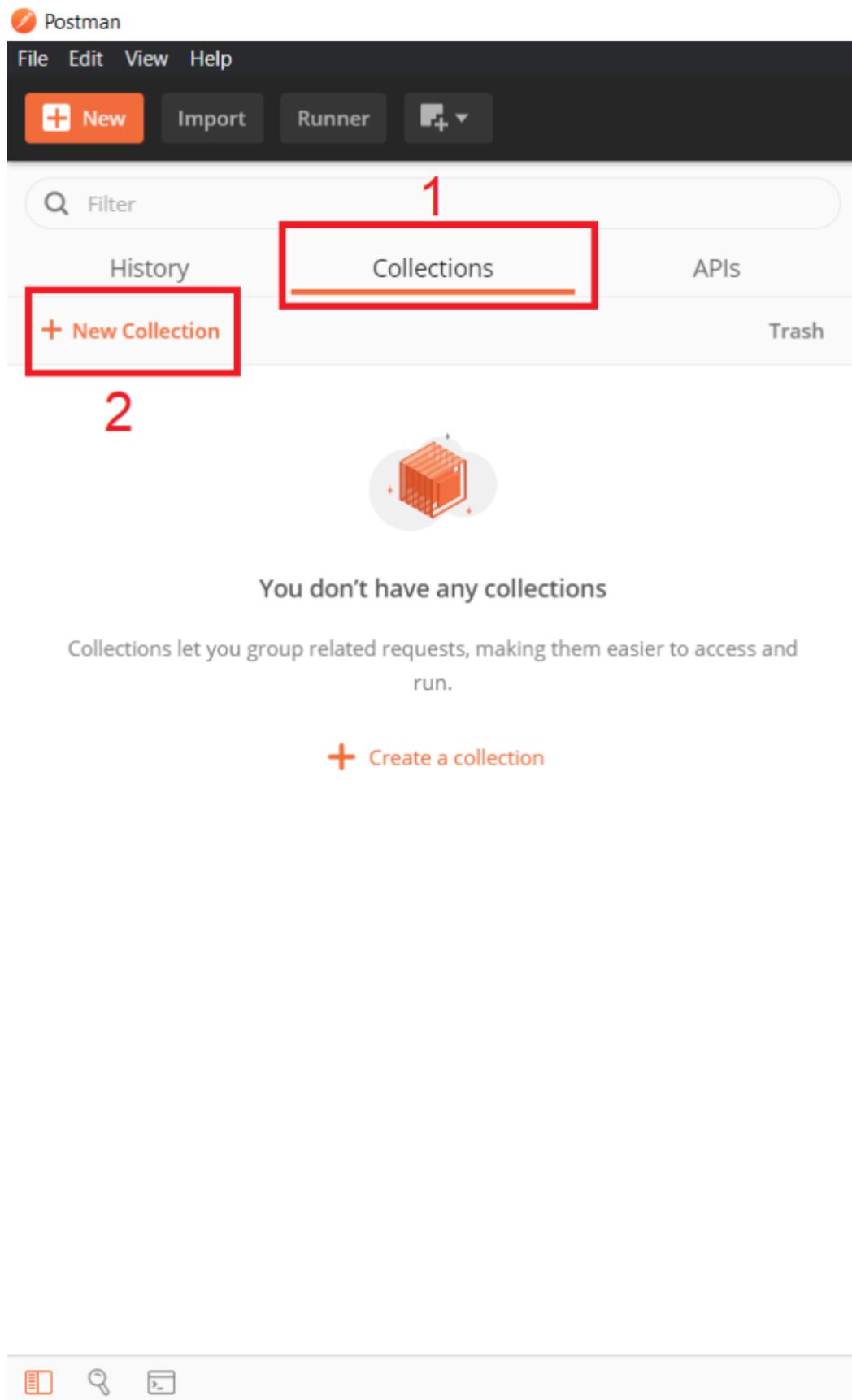
Postman merupakan salah satu aplikasi terpopuler dalam menangani API. Untuk menggunakannya Anda bisa terlebih dahulu mendownload nya di link berikut: <https://www.postman.com/downloads/>. Setelah mendownload aplikasi Postman tersebut, lakukan instalasi dan kemudian lakukan Sign Up (bisa dilakukan melalui aplikasi atau melalui web berikut: <https://identity.getpostman.com/signup>). Setelah Sign Up dan berhasil masuk ke dalam aplikasi

Postman, berikut tampilan yang akan Anda temukan:



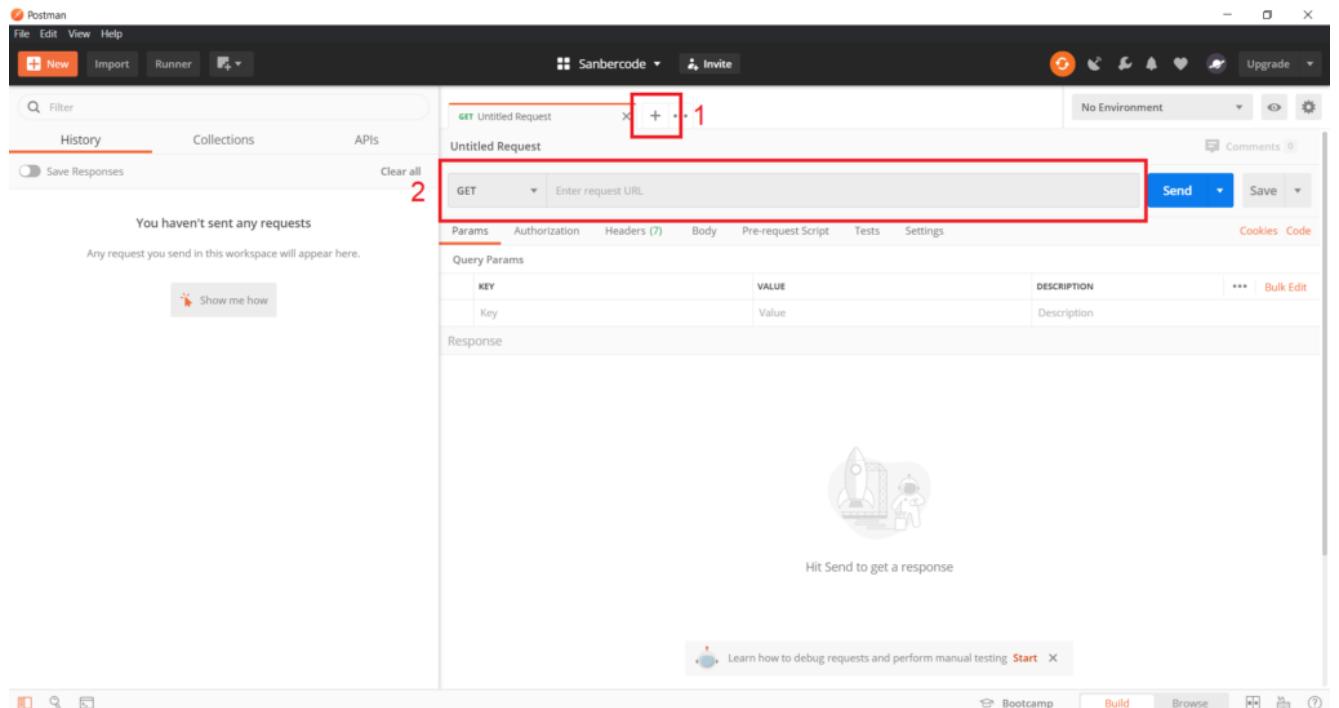
Collection

Disarankan untuk mengelompokkan API yang akan digunakan pada satu Collection, dengan cara memilih menu “Collection” pada sidebar dan klik “+ New Collection” kemudian memasukkan nama Collection tersebut, bisa menggunakan nama Project maupun nama web / sumber API nya.



Request

Selanjutnya Anda bisa menambahkan request API dengan menekan tombol “+” pada menu yang tersedia (1) . Selanjutnya Anda bisa memasukkan URL yang dituju pada bagian yang telah disediakan (2). Pada Postman telah disediakan berbagai method yang dapat Anda temukan pada sebelah kiri dari kolom input URL, mulai dari GET, POST, PUT, DELETE dan lainnya.



Dibawah kolom input URL, Anda akan menemukan menu/tab “Params”, “Authentication”, “Header”, “Body”, dan lainnya.

- Menu “Params” yang dapat digunakan untuk menambah query yang akan membuat URL yang digunakan bertambah panjang (biasanya diawali dengan “?”).
- Pada menu “Authentication”, Anda akan menemukan metode Auth yang terkadang diperlukan untuk dapat mengakses suatu API (umumnya menggunakan API Key, Bearer Token, atau OAuth tergantung sumber/penyedia API).
- Pada menu “Header”, Anda juga dapat mengatur Authentication dengan menggunakan key “Authentication” dan value dari Authentication yang diperlukan (misal bearer {token}). Pada bagian ini biasanya dimasukkan informasi tambahan yang digunakan oleh back-end/server untuk mengatur bagaimana cara response akan diberikan.

- Menu “Body”, yang biasa diperlukan ketika menggunakan method “POST” atau mengirim suatu form kepada back-end/server.

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

1. Mengimplementasi REST API menggunakan api dari online untuk method nya ada 2 berikut link untuk implementasinya

- **Get untuk meminta data dari server**

GET: <https://reqres.in/api/users?page=2>

- **Get Detail untuk mengembalikan data dari local ke server**

- **Get Detail:** <https://reqres.in/api/users/:id>

2. Membuka Postman

3. Masukkan url, pilih method get setting header sehingga seperti berikut ini:

The screenshot shows the Postman interface with a red box highlighting the 'GET' method and the URL 'https://reqres.in/api/users?page=2'. Below this, the 'Headers' tab is selected, showing a table with one row: 'Accept' set to 'application/json'. The 'Body' tab is also visible at the bottom.

| KEY | VALUE |
|--------|------------------|
| Accept | application/json |

4. Maka akan muncul data Json

```
{
  "page": 2,
```

```
"per_page": 6,
"total": 12,
"total_pages": 2,
"data": [
  {
    "id": 7,
    "email": "michael.lawson@reqres.in",
    "first_name": "Michael",
    "last_name": "Lawson",
    "avatar": "https://reqres.in/img/faces/7-image.jpg"
  },
  {
    "id": 8,
    "email": "lindsay.ferguson@reqres.in",
    "first_name": "Lindsay",
    "last_name": "Ferguson",
    "avatar": "https://reqres.in/img/faces/8-image.jpg"
  },
  {
    "id": 9,
    "email": "tobias.funke@reqres.in",
    "first_name": "Tobias",
    "last_name": "Funke",
    "avatar": "https://reqres.in/img/faces/9-image.jpg"
  },
  {
    "id": 10,
    "email": "byron.fields@reqres.in",
    "first_name": "Byron",
    "last_name": "Fields",
    "avatar": "https://reqres.in/img/faces/10-image.jpg"
  },
  {
    "id": 11,
    "email": "george.edwards@reqres.in",
    "first_name": "George",
    "last_name": "Edwards",
    "avatar": "https://reqres.in/img/faces/11-image.jpg"
  },
  {
    "id": 12,
    "email": "rachel.howell@reqres.in",
    "first_name": "Rachel",
    "last_name": "Howell",
    "avatar": "https://reqres.in/img/faces/12-image.jpg"
  }
]
```

```

        }
    ],
    "support": {
        "url": "https://reqres.in/#support-heading",
        "text": "To keep ReqRes free, contributions towards server costs are appreciated!"
    }
}

```

5. Masukkan url untuk mendapatkan detail dari salah satu data, pilih method get setting header sehingga seperti berikut ini:

The screenshot shows the Postman interface with the following configuration:

- Method:** GET
- URL:** https://reqres.in/api/users/11
- Headers (8):**
 - Accept: application/json
- Body:** (Empty)
- Cookies:** (Empty)
- Tests:** (Empty)
- Settings:** (Empty)

The "Accept" header row is highlighted with a red box. The JSON response body is shown below:

```

1 {
2     "data": {
3         "id": 11,
4         "email": "george.edwards@reqres.in",
5         "first_name": "George",
6         "last_name": "Edwards",
7         "avatar": "https://reqres.in/img/faces/11-image.jpg"
8     },
9     "support": {
10        "url": "https://reqres.in/#support-heading",
11        "text": "To keep ReqRes free, contributions towards server costs are appreciated!"
12    }
13 }

```

6. Maka akan muncul data Json

```

{
    "data": {
        "id": 11,
        "email": "george.edwards@reqres.in",
        "first_name": "George",
        "last_name": "Edwards",
        "avatar": "https://reqres.in/img/faces/11-image.jpg"
    },
    "support": {
        "url": "https://reqres.in/#support-heading",
        "text": "To keep ReqRes free, contributions towards server costs are appreciated!"
    }
}

```

f. Hasil dan Pembahasan

1. Dokumentasi tugas berupa laporan secara individu dengan mencari source API yang berbeda dengan yang ada di prosedur kerja kemudian dikumpulkan dalam bentuk laporan PDF dan dikumpulkan di LMS.

g. Kesimpulan

Mahasiswa memahami konsep dasar Rest API dengan mampu memahami setiap baris code dari implementasi Rest API.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|--------------------|--|-------|---|---|---|
| | | 1 | 2 | 3 | 4 |
| 1 | Ketepatan waktu dan ketepatan implementasi pada konsep Rest API | 1 | 2 | 3 | 4 |
| 2 | Ketepatan waktu dan ketepatan implementasi pada konsep Rest API | 1 | 2 | 3 | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat implementasi pada konsep Rest API | 1 | 2 | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan implementasi pada konsep Rest API | 1 | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 34 : REST API

| | |
|----------------------------------|---|
| Pokok Bahasan | : Rest API |
| Acara Praktikum/Pertemuan | : Minggu 10 / 2 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami Rest API
2. Mahasiswa mampu implementasi Rest API.

b. Indikator

Keberhasilan dalam memahami Rest API, serta implementasi code Rest API.

c. Dasar Teori

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

1. Membuat project dengan nama minggu_10
2. Tambahkan library pada file pubspec.yaml seperti berikut:

```
23 dependencies:  
24   flutter:  
25     | sdk: flutter  
26  
27  
28   # The following adds the Cupertino Icons font to your application.  
29   # Use with the CupertinoIcons class for iOS style icons.  
30   cupertino_icons: ^1.0.2  
31   http: ^0.13.3
```

3. Membuat file get_detail_screen.dart yang berisi

```

1 import 'dart:convert';
2 
3 import 'package:flutter/material.dart';
4 import 'package:http/http.dart' as http;
5 
6 class GetDataDetailScreen extends StatefulWidget {
7   int? value;
8   GetDataDetailScreen({Key? key, this.value}) : super(key: key);
9 
10  @override
11  _GetDataDetailScreenState createState() => _GetDataDetailScreenState(value);
12 }
13 
14 class _GetDataDetailScreenState extends State<GetDataDetailScreen> {
15   int? value;
16   _GetDataDetailScreenState(this.value);
17   Map? data;
18   String? uri;
19   @override
20   void initState() {
21     var url = "https://reqres.in/api/users/${value.toString()}";
22     _getRefreshDaata(url);
23 
24     print("susu +${value}");
25   }
26 
27 
28   Future<void> _getRefreshDaata(url) async {
29     getJsonData(context, url);
30   }
31 
32   Future<void> getJsonData(BuildContext context, url) async {
33     setState(() {
34       uri = url;
35     });
36     var response = await http
37       .get(Uri.parse(uri.toString()), headers: {"Accept": "application/json"});
38     print(response.body);
39     setState(() {
40       var convertDataToJson = jsonDecode(response.body);
41       data = convertDataToJson['data'];
42     });
43   }
44 
45   @override
46   Widget build(BuildContext context) {

```

```

47    return Scaffold(
48        appBar: AppBar(
49            title: Text("Get data api regres"),
50        ), // AppBar
51        body: Container(
52            child: data == null
53                ? Center(
54                    child: Container(
55                        width: 100,
56                        height: 100,
57                        decoration: BoxDecoration(
58                            borderRadius: BorderRadius.circular(6),
59                            color: Colors.grey), // BoxDecoration
60                        child: Column(
61                            mainAxisAlignment: MainAxisAlignment.center,
62                            children: [
63                                CircularProgressIndicator(),
64                                SizedBox(
65                                    height: 10,
66                                ), // SizedBox
67                                Text(
68                                    "Loading . . .",
69                                    style: TextStyle(color: Colors.white),
70                                ) // Text
71                            ],
72                        )), // Column // Container // Center
73                    : Container(
74                        child: ListTile(
75                            leading: Image.network(data!["avatar"]),
76                            title:
77                                Text(data!["first_name"] + " " + data!["last_name"]),
78                            subtitle: Text(data!["email"]),
79                        ), // ListTile
80                    )), // Container // Container // Scaffold
81    }
82 }

```

4. Membuat file get_data.dart yang berisi

```

1 import 'dart:convert';
2 import 'package:flutter/material.dart';
3 import 'package:http/http.dart' as http;
4 import 'package:minggu_10/get_detail_screen.dart';
5
6 class GetDataScreen extends StatefulWidget {
7   const GetDataScreen({Key? key}) : super(key: key);
8
9   @override
10  _GetDataScreenState createState() => _GetDataScreenState();
11 }
12
13 class _GetDataScreenState extends State<GetDataScreen> {
14   final String url = "https://reqres.in/api/users?page=2";
15   List? data;
16   @override
17   void initState() {
18     _getRefreshDaata();
19     super.initState();
20   }
21
22   Future<void> _getRefreshDaata() async {
23     this.getJsonData(context);
24   }
25
26   Future<void> getJsonData(BuildContext context) async {
27     var response = await http.get(Uri.parse(url), headers: {
28       "Accept": "application/json",
29     });
30     print(response.body);
31     setState(() {
32       var convertDataToJson = jsonDecode(response.body);
33       data = convertDataToJson['data'];
34     });
35   }
36
37   @override
38   Widget build(BuildContext context) {
39     return Scaffold(
40       appBar: AppBar(
41         title: Text("Get data api regres"),
42       ), // AppBar
43       body: RefreshIndicator(
44         onRefresh: _getRefreshDaata,
45         child: data == null
46           ? Center(child: CircularProgressIndicator())
47           : ListView.builder(
48             itemCount: data!.length,
49             itemBuilder: (context, index) {
50               return Card(
51                 margin: EdgeInsets.all(10),
52                 child: Padding(
53                   padding: EdgeInsets.all(10),
54                   child: Column(
55                     mainAxisAlignment: MainAxisAlignment.start,
56                     crossAxisAlignment: CrossAxisAlignment.start,
57                     children: [
58                       Text("Name: ${data![index]['name']}"),
59                     SizedBox(height: 10),
60                     Text("Email: ${data![index]['email']}"),
61                     SizedBox(height: 10),
62                     Text("Avatar: ${data![index]['avatar']}"),
63                   ],
64                 ),
65               );
66             },
67           )
68         );
69       )
70     );
71   }
72 }

```

```
46 |             ? Center(child: CircularProgressIndicator())
47 |             : ListView.builder(
48 |                 itemCount: data == null ? 0 : data!.length,
49 |                 itemBuilder: (BuildContext context, int index) {
50 |                     return Container(
51 |                         padding: EdgeInsets.all(5.0),
52 |                         child: Column(
53 |                             children: [
54 |                                 GestureDetector(
55 |                                     onTap: () {
56 |                                         Navigator.push(
57 |                                             context,
58 |                                             MaterialPageRoute(
59 |                                                 builder: (context) => GetDataDetailScreen(
60 |                                                     value: data![index]["id"]),
61 |                                             );
62 |                                         },
63 |                                         child: Padding(
64 |                                             padding: EdgeInsets.all(16.0),
65 |                                             child: Row(
66 |                                                 children: [
67 |                                                     ClipRRect(
68 |                                                         child: Image.network(
69 |                                                             data![index]["avatar"],
70 |                                                             height: 80,
71 |                                                             width: 80,
72 |                                                         ),
73 |                                                         ),
74 |                                                     SizedBox(width: 10),
75 |                                                     Column(
76 |                                                         children: [
77 |                                                             Text(data![index]["first_name"] +
78 |                                                                 " " +
79 |                                                                 data![index]["last_name"]),
80 |                                                             Text(data![index]["email"]),
81 |                                                         ],
82 |                                                     ),
83 |                                                 ],
84 |                                             ),
85 |                                         ),
86 |                                         ),
87 |                                         Divider()
```

```

88     ],
89     ),
90   );
91 });
92 );
93 }
94 }

```

5. Membuat file main.dart yang berisi

```

4 void main() {
5   runApp(
6     MyApp(),
7   );
8 }
9
10 class MyApp extends StatelessWidget {
11   // This widget is the root of your application.
12   @override
13   Widget build(BuildContext context) {
14     return MaterialApp(
15       debugShowCheckedModeBanner: false,
16       title: 'Get API',
17       theme: ThemeData(
18         primarySwatch: Colors.blue,
19       ),
20       // home: PageOne(),
21       // getPages: pageRouteApp.pages,
22       home: GetDataScreen(),
23     );
24   }
25 }

```

f. Hasil dan Pembahasan

1. Dokumentasi tugas berupa laporan secara individu penjelasan pada prosedur kerja kemudian dikumpulkan dalam bentuk laporan PDF dan dikumpulkan di LMS.
2. Pengumpulan source code melalui link commit gitlab/github pada LMS

g. Kesimpulan

Mahasiswa mendapatkan memahami Rest API, serta implementasi code Rest API.

h. Rubrik Penilaian

| No | Indikator | Skor* |
|----|---|----------------|
| 1 | Ketepatan waktu dan ketepatan implementasi pada konsep Rest API | 1 2 3 4 |

| | | | | | |
|--------------------|--|-----|-----|-----|---|
| 2 | Ketepatan waktu dan ketepatan implementasi pada konsep Rest API | 1 | 2 | (3) | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat implementasi pada konsep Rest API | 1 | (2) | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan implementasi pada konsep Rest API | (1) | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 35 : REST API

| | |
|----------------------------------|--|
| Pokok Bahasan | : Pengaturan Android Manifest, Asset dan <i>Activity</i> |
| Acara Praktikum/Pertemuan | : Minggu 10 / 3 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

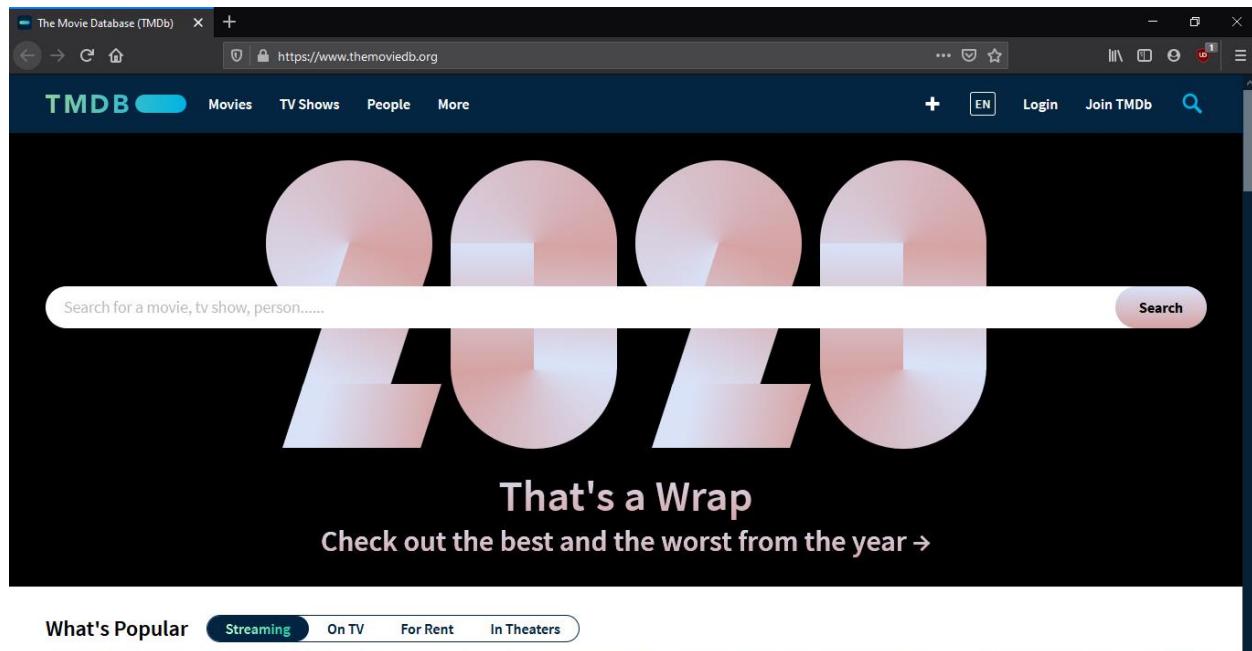
1. Mahasiswa mampu memahami Rest API
2. Mahasiswa mampu implementasi Rest API.

b. Indikator

Keberhasilan dalam memahami Rest API, serta implementasi code Rest API.

c. Dasar Teori

Sebuah website bernama The Movie DB, website penyedia informasi tentang film dan juga menyediakan API secara gratis dengan syarat harus register terlebih duluh.



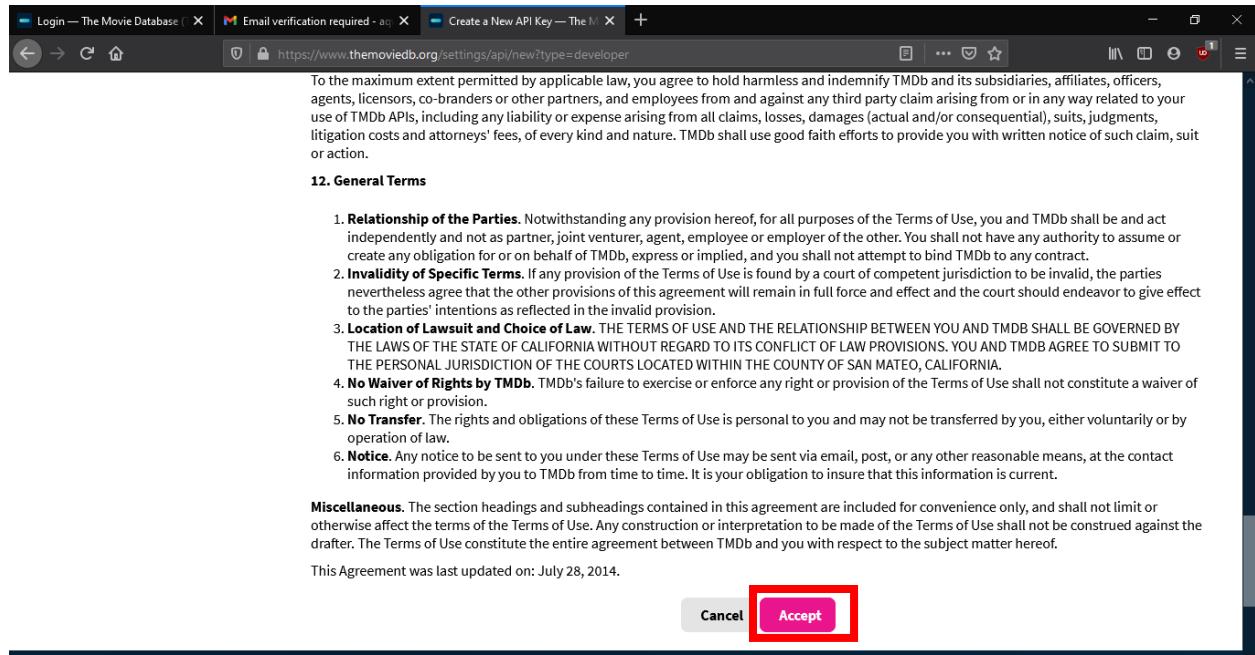
Untuk mendapatkan API, masuk ke website tersebut, lalu pilih Sign Up. Selanjutnya masuk ke akun, Untuk mengakses API-nya diperlukan API key, terlebih dahulu harus buat akun. Setelah itu akses ke menu settings di profile, kemudian generate API key di menu API, jika sudah maka akan tampil data seperti berikut.

The screenshot shows the TMDB profile page for user 'aqwuarvitha'. At the top, there's a navigation bar with 'TMDB' and links for 'Movies', 'TV Shows', 'People', and 'More'. Below the navigation is a large circular profile picture with a blue and white power symbol. The user's name 'aqwuarvitha' is displayed next to it, along with the text 'Member since February 2021'. Two circular icons show '0x' average movie score and '0x' average TV score. A dropdown menu on the right includes options like 'View profile', 'Discussions', 'Lists', 'Ratings', 'Watchlist', 'Edit Profile', and 'Logout'. A red box highlights the 'Settings' option in the dropdown. Below the profile section, there's a 'Stats' summary with 'Total Edits' (0) and 'Total Ratings' (0). To the right is a 'Rating Overview' chart showing a distribution from 1 to 10. A section titled 'Upcoming From Watchlist' indicates no upcoming movies. A 'Recent Discussions' section is also present.

The screenshot shows the 'My Settings' page at <https://www.themoviedb.org/settings/account>. On the left, a sidebar titled 'Settings' contains links for 'Edit Profile', 'Account Settings' (which is highlighted in pink), 'Streaming Services', 'Notification Settings', 'Blocked Users', 'Import List', 'Sharing Settings', 'Social Settings', 'Sessions' (with 'API' highlighted in a red box), and 'Delete Account'. The main content area is titled 'Account Settings' and includes fields for 'Default Language' (set to English (en-US)), 'Fallback Language' (set to None (Don't Fallback)), 'Country' (set to Indonesia), 'Timezone - Auto detect?' (checked, set to Asia - Bangkok), 'Include Adult Items in Search?' (set to No), and 'Filter Profanity?' (set to Yes). A 'Save' button is at the bottom. A URL at the bottom of the page is <https://www.themoviedb.org/settings/api>.

The screenshot shows the TMDB API Settings page. The URL is <https://www.themoviedb.org/settings/api>. The 'Create' tab is selected. On the left, there's a sidebar with 'Settings' and various options like Edit Profile, Account Settings, etc. The main content area has sections for Documentation and Support, both with links to external sites. A red box highlights the 'Request an API Key' button, which contains the text 'To generate a new API key, click here.'

The screenshot shows the 'Create a New API Key' page. The URL is <https://www.themoviedb.org/settings/api/request>. It asks 'What type of API key do you wish to register?' with two options: 'Developer' and 'Professional'. The 'Developer' option is highlighted with a red box. The 'Developer' description includes: 'You are an individual', 'Your project is still in development', 'Your project is non profit', and 'Your project is ad supported'. The 'Professional' description includes: 'You represent a company', 'Your project is for profit (not ad supported)', and 'You are an OEM or hardware vendor'. Below the page is a link: <https://www.themoviedb.org/settings/api/new?type=developer>.



TMDB Movies TV Shows People More

aqwuarvitha

Settings

API Overview Create

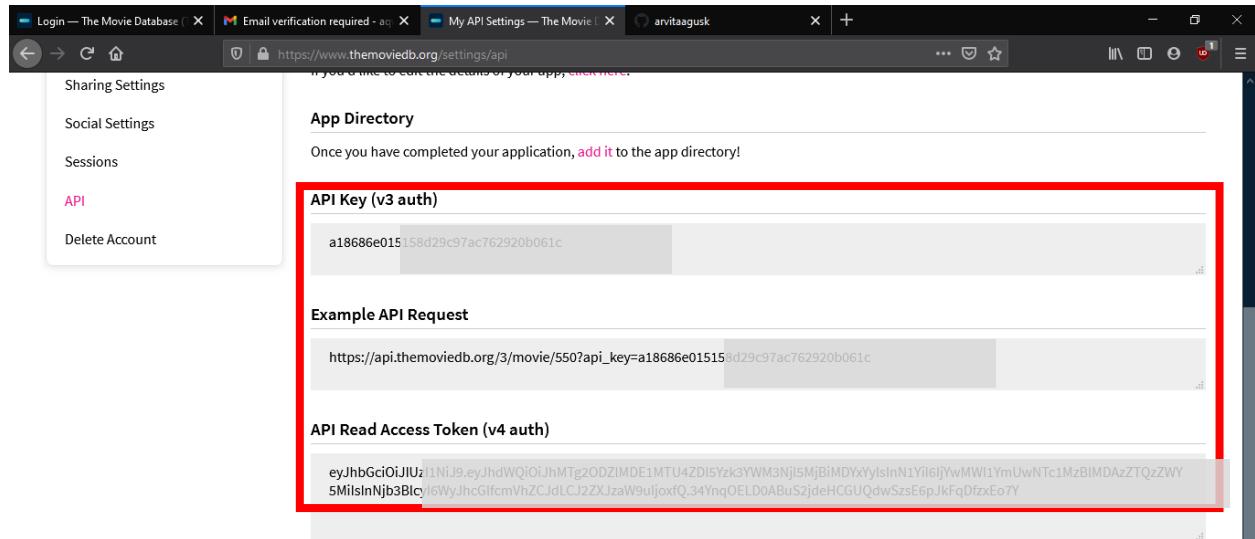
Type of Use
Mobile Application

Application Name
MovieDB

Application URL

Application Summary

**Application URL dapat menggunakan url github project*



d. Alat dan Bahana

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

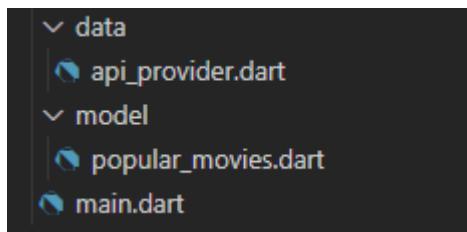
1. Membuat project dengan nama minggu_10_moviedb
2. Tambahkan library pada file pubspec.yaml seperti berikut:

```

23 dependencies:
24   flutter:
25     sdk: flutter
26   cached_network_image: 2.0.0-rc
27   http:
28
29   # The following adds the Cupertino Icons font to your application.
30   # Use with the CupertinoIcons class for iOS style icons.
31   cupertino_icons: ^1.0.2
32

```

3. Membuat folder dan file seperti berikut:



4. Membuat file popular_movies.dart pada lib/model yang berisi class PopularMovies dan Results

- PopularMovies

```
1  class PopularMovies {  
2      late int page;  
3      late int totalResults;  
4      late int totalPages;  
5      late List<Results> results;  
6  
7      PopularMovies(  
8          {required this.page,  
9           required this.totalResults,  
10          required this.totalPages,  
11           required this.results});  
12  
13      PopularMovies.fromJson(Map<String, dynamic> json) {  
14          page = json['page'];  
15          totalResults = json['total_results'];  
16          totalPages = json['total_pages'];  
17          if (json['results'] != null) {  
18              results = <Results>[];  
19              json['results'].forEach((v) {  
20                  results.add(new Results.fromJson(v));  
21              });  
22          }  
23      }  
24      Map<String, dynamic> toJson() {  
25          final Map<String, dynamic> data = new Map<String, dynamic>();  
26          data['page'] = this.page;  
27          data['total_results'] = this.totalResults;  
28          data['total_pages'] = this.totalPages;  
29          if (this.results != null) {  
30              data['results'] = this.results.map((v) => v.toJson()).toList();  
31          }  
32          return data;  
33      }  
34  }
```

- Results

```
37 class Results {  
38     late double popularity;  
39     late int voteCount;  
40     late bool video;  
41     late String posterPath;  
42     late int id;  
43     late bool adult;  
44     late String backdropPath;  
45     late String originalLanguage;  
46     late String originalTitle;  
47     late List<int> genreIds;  
48     late String title;  
49     late double voteAverage;  
50     late String overview;  
51     late String releaseDate;  
52  
53     Results(  
54         {required this.popularity,  
55         required this.voteCount,  
56         required this.video,  
57         required this.posterPath,  
58         required this.id,  
59         required this.adult,  
60         required this.backdropPath,  
61         required this.originalLanguage,  
62         required this.originalTitle,  
63         required this.genreIds,  
64         required this.title,  
65         required this.voteAverage,  
66         required this.overview,  
67         required this.releaseDate});  
68  
69     Results.fromJson(Map<String, dynamic> json) {  
70         popularity = json['popularity'];  
71         voteCount = json['vote_count'];  
72         video = json['video'];  
73         posterPath = json['poster_path'];  
74         id = json['id'];  
75         adult = json['adult'];  
76         backdropPath = json['backdrop_path'];  
77         originalLanguage = json['original_language'];  
78         originalTitle = json['original_title'];
```

```

79     genreIds = json['genre_ids'].cast<int>();
80     title = json['title'];
81     voteAverage = json['vote_average'].toDouble();
82     overview = json['overview'];
83     releaseDate = json['release_date'];
84   }
85
86   Map<String, dynamic> toJson() {
87     final Map<String, dynamic> data = new Map<String, dynamic>();
88     data['popularity'] = this.popularity;
89     data['vote_count'] = this.voteCount;
90     data['video'] = this.video;
91     data['poster_path'] = this.posterPath;
92     data['id'] = this.id;
93     data['adult'] = this.adult;
94     data['backdrop_path'] = this.backdropPath;
95     data['original_language'] = this.originalLanguage;
96     data['original_title'] = this.originalTitle;
97     data['genre_ids'] = this.genreIds;
98     data['title'] = this.title;
99     data['vote_average'] = this.voteAverage;
100    data['overview'] = this.overview;
101    data['release_date'] = this.releaseDate;
102    return data;
103  }
104}

```

5. Membuat file api_provider.dart pada folder lib/data sebagai berikut

```

1 import 'dart:convert';
2 import 'package:http/http.dart' show Client, Response;
3 import 'package:minggu_10_moviedb/model/popular_movies.dart';
4
5 class ApiProvider {
6   String apiKey = '0fc5740199faa752da813c8c97f659e8';
7   String baseUrl = 'https://api.themoviedb.org/3';
8
9   Client client = Client();
10
11  Future<PopularMovies> getPopularMovies() async {
12    // String url = '$baseUrl/movie/popular?api_key=$apiKey';
13    // print(url);
14    Response response =
15      await client.get('$baseUrl/movie/popular?api_key=$apiKey');
16

```

```

17     if (response.statusCode == 200) {
18         return PopularMovies.fromJson(jsonDecode(response.body));
19     } else {
20         throw Exception(response.statusCode);
21     }
22 }
23 }
```

6. Membuat file main.dart sebagai berikut:

```

1 import 'package:cached_network_image/cached_network_image.dart';
2 import 'package:flutter/material.dart';
3
4 import 'data/api_provider.dart';
5 import 'model/popular_movies.dart';
6
7 Run | Debug | Profile
8 void main() => runApp(MoviesApp());
9
10 class MoviesApp extends StatelessWidget {
11     @override
12     Widget build(BuildContext context) {
13         return MaterialApp(
14             title: 'Movies App',
15             theme: ThemeData(
16                 primarySwatch: Colors.indigo,
17             ), // ThemeData
18             home: Home(),
19         ); // MaterialApp
20     }
21 }
```

```

22 class Home extends StatefulWidget {
23     @override
24     _HomeState createState() => _HomeState();
25 }
26
27 class _HomeState extends State<Home> {
28     ApiProvider apiProvider = ApiProvider();
29     late Future<PopularMovies> popularMovies;
30
31     String imageBaseUrl = 'https://image.tmdb.org/t/p/w500';
32 }
```

```
33  @override
34  void initState() {
35      popularMovies = apiProvider.getPopularMovies();
36      super.initState();
37  }
38
39  @override
40  Widget build(BuildContext context) {
41      return Scaffold(
42          appBar: AppBar(
43              title: Text('Movies App'),
44          ), // AppBar
45          body: FutureBuilder(
46              future: popularMovies,
47              builder: (BuildContext context, AsyncSnapshot snapshot) {
48                  if (snapshot.hasData) {
49                      print("Has Data: ${snapshot.hasData}");
50                      return ListView.builder(
51                          itemCount: snapshot.data.results.length,
52                          itemBuilder: (BuildContext context, int index) {
53                              return moviesItem(
54                                  poster:
55                                      '$imageBaseUrl${snapshot.data.results[index].posterPath}',
56                                      title: '${snapshot.data.results[index].title}',
57                                      date: '${snapshot.data.results[index].releaseDate}',
58                                      voteAverage: '${snapshot.data.results[index].voteAverage}',
59                                      onTap: () {
60                                          Navigator.of(context).push(MaterialPageRoute(
61                                              builder: (context) => MovieDetail(
62                                                  movie: snapshot.data.results[index],
63                                              ))); // MovieDetail // MaterialPageRoute
64                                      });
65                  },
66              ); // ListView.builder
67          } else if (snapshot.hasError) {
68              print("Has Error: ${snapshot.hasError}");
69              return Text('Error!!!');
70          } else {
71              print("Loading...");
72              return CircularProgressIndicator();
73          }
74      },
75  ), // FutureBuilder
```

```
76    ); // Scaffold
77 }
78
79 Widget moviesItem(
80   required String poster,
81   required String title,
82   required String date,
83   required String voteAverage,
84   required Function()? onTap}) {
85   return InkWell(
86     onTap: onTap,
87     child: Container(
88       margin: EdgeInsets.all(10),
89       child: Card(
90         child: Container(
91           child: Row(
92             mainAxisAlignment: MainAxisAlignment.start,
93             children: <Widget>[
94               Container(
95                 width: 120,
96                 child: CachedNetworkImage(
97                   imageUrl: poster,
98                 ), // CachedNetworkImage
99               ), // Container
100              SizedBox(
101                width: 20,
102              ), // SizedBox
103              Expanded(
104                child: Container(
105                  margin: EdgeInsets.only(top: 20),
106                  child: Column(
107                    mainAxisAlignment: MainAxisAlignment.start,
108                    mainAxisSize: MainAxisSize.min,
109                    children: <Widget>[
110                      Text(
111                        title,
112                        style: TextStyle(
113                          fontSize: 18, fontWeight: FontWeight.w600), // Text
114                      ), // Text
115                      SizedBox(
116                        height: 10,
```

```
117     ), // SizedBox
118     Row(
119       children: <Widget>[
120         Icon(
121           Icons.calendar_today,
122           size: 12,
123         ), // Icon
124         SizedBox(
125           width: 5,
126         ), // SizedBox
127         Text(date),
128       ], // <Widget>[]
129     ), // Row
130     SizedBox(
131       height: 10,
132     ), // SizedBox
133     Row(
134       children: <Widget>[
135         Icon(
136           Icons.star,
137           size: 12,
138         ), // Icon
139         SizedBox(
140           width: 5,
141         ), // SizedBox
142         Text(voteAverage),
143       ], // <Widget>[]
144     ), // Row
145   ], // <Widget>[]
146   ), // Column
147   ), // Container
148   ), // Expanded
149   ], // <Widget>[]
150   ), // Row
151   ), // Container
152   ), // Card
153   ), // Container
154 ); // InkWell
155 }
156 }
```

```

158 class MovieDetail extends StatelessWidget {
159   final Results movie;
160
161   const MovieDetail({Key? key, required this.movie}) : super(key: key);
162
163   @override
164   Widget build(BuildContext context) {
165     return Scaffold(
166       appBar: AppBar(
167         title: Text(movie.title),
168       ), // AppBar
169       body: Container(
170         child: Text(movie.overview),
171       ), // Container
172     ); // Scaffold
173   }
174 }
```

f. Hasil dan Pembahasan

1. Dokumentasi tugas berupa laporan secara individu penjelasan pada prosedur kerja kemudian dikumpulkan dalam bentuk laporan PDF dan dikumpulkan di LMS.
2. Pengumpulan source code melalui link commit gitlab/github pada LMS

g. Kesimpulan

Mahasiswa mendapatkan memahami Rest API, serta implementasi code Rest API.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|--------------------|--|-------|-----|-----|-----|
| 1 | Ketepatan waktu dan ketepatan implementasi pada konsep Rest API | 1 | 2 | 3 | (4) |
| 2 | Ketepatan waktu dan ketepatan implementasi pada konsep Rest API | 1 | 2 | (3) | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat implementasi pada konsep Rest API | 1 | (2) | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan implementasi pada konsep Rest API | (1) | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 36 : REST API

| | |
|----------------------------------|--|
| Pokok Bahasan | : Menangkap Perubahan Nilai dari Suatu Sensor |
| Acara Praktikum/Pertemuan | : Minggu 10 / 4 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami Rest API
2. Mahasiswa mampu implementasi Rest API

b. Indikator

Keberhasilan dalam memahami Rest API, serta implementasi code Rest API.

c. Dasar Teori

-

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

1. Membuat aplikasi Rest API sederhana berdasarkan API yang telah didapatkan pada Acara 33

f. Hasil dan Pembahasan

1. Dokumentasi tugas berupa laporan secara individu penjelasan pada prosedur kerja kemudian dikumpulkan dalam bentuk laporan PDF dan dikumpulkan di LMS.
2. Pengumpulan source code melalui link commit gitlab/github pada LMS

g. Kesimpulan

Mahasiswa mendapatkan memahami Rest API, serta implementasi code Rest API.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|--------------------|--|-------|-----|-----|-----|
| 1 | Ketepatan waktu dan ketepatan implementasi pada konsep Rest API | 1 | 2 | 3 | (4) |
| 2 | Ketepatan waktu dan ketepatan implementasi pada konsep Rest API | 1 | 2 | (3) | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat implementasi pada konsep Rest API | 1 | (2) | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan implementasi pada konsep Rest API | (1) | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 37 : AUTHENTICATION

| | |
|----------------------------------|---|
| Pokok Bahasan | : Authentication |
| Acara Praktikum/Pertemuan | : Minggu 11 / 1 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

- a. Mahasiswa mampu memahami konsep Authentication.
- b. Mahasiswa mampu menerapkan Authentication.

b. Indikator

Keberhasilan mahasiswa memahami konsep Authentication dan menerapkan Authentication pada flutter.

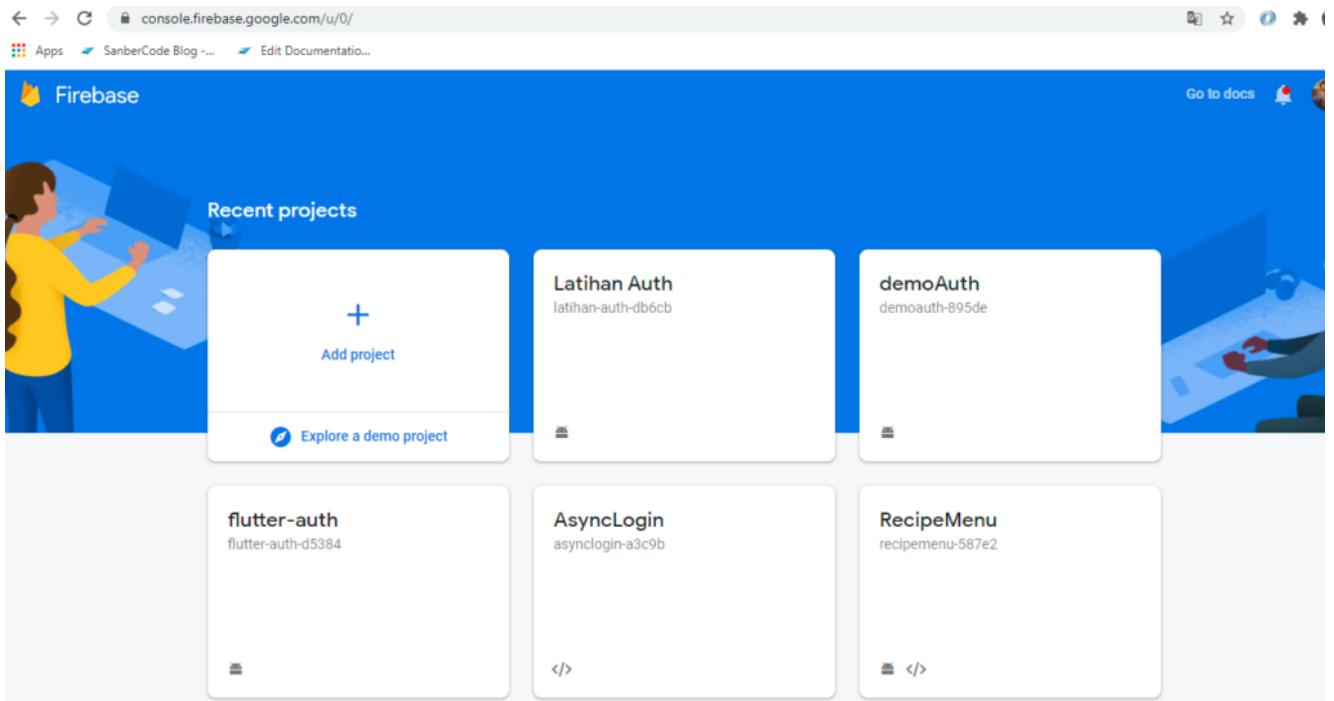
c. Dasar Teori

Selain menggunakan API yang telah disediakan back-end developer maupun oleh pihak ketiga (third party), Anda juga dapat membuat API sendiri, misalnya dalam bentuk database dengan menggunakan Firebase. Authentication adalah metode keamanan yang diterapkan saat pengguna akan memasuki sebuah sistem atau aplikasi dengan cara harus memasukan email/username dan password untuk memastikan bahwa pengguna memang memiliki hak akses ke sebuah sistem.

Untuk mulai menggunakannya Anda cukup menuju <https://firebase.google.com/> dan login menggunakan akun Google Anda. Langkah-langkah yang diperlukan selanjutnya adalah sebagai berikut:

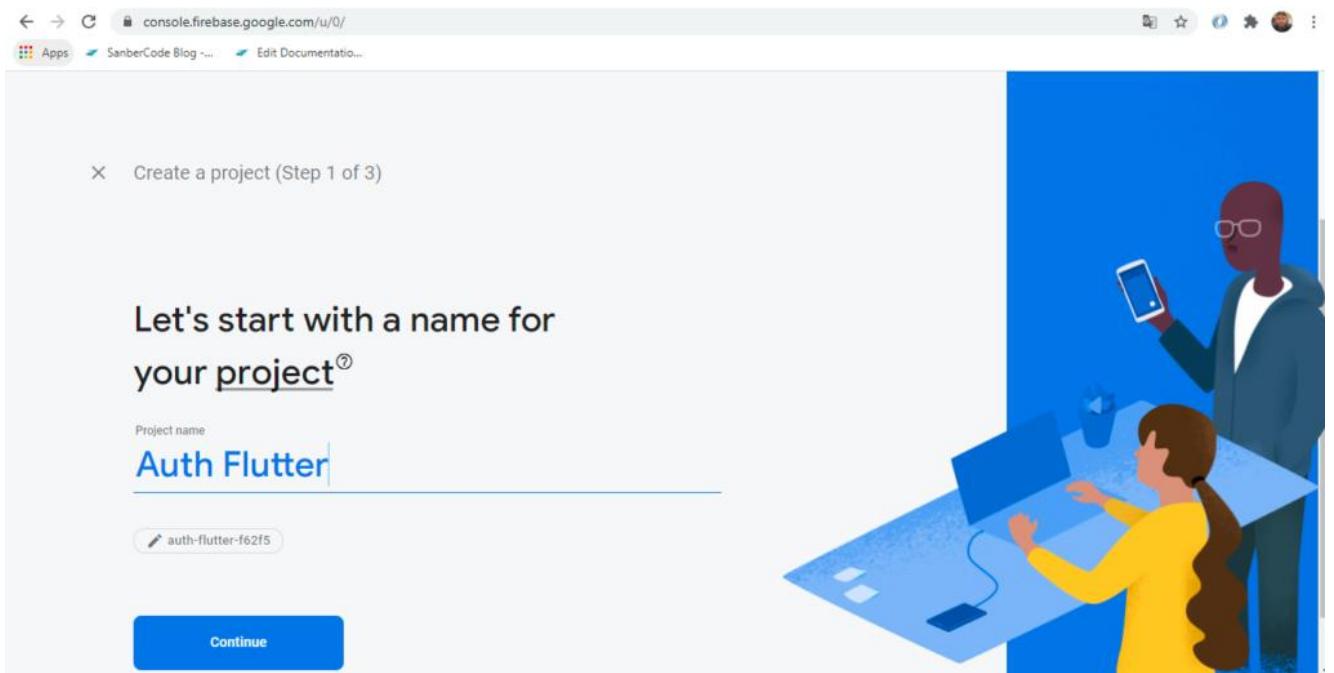
1. Klik “Go to Console” yang berada pada bagian kanan atas.
2. Buat Project baru dengan klik “Create Project” atau “Add Project” jika sudah pernah membuat project sebelumnya.
3. Masukkan nama project Anda dan klik “Continue”.
4. Anda bisa meng-enable “Google Analytics” dan klik “Continue”.
5. Kemudian Anda akan diminta untuk memilih akun yang digunakan untuk menerima data dari Google Analytics, pilih “Default Account for Firebase” saja, selanjutnya klik “Create Project”.

6. Atau Anda bisa meng-disable “Google Analytic” dan klik “Create Project”.
7. Selanjutnya Anda tinggal menunggu project selesai dibuat.



tampilan awal firebase

untuk pembuatan project klik tombol tambah



setelah add akan muncul seperti ini dan bisa di continue

The screenshot shows the Firebase Project Overview page for a project named 'Auth Flutter'. The left sidebar lists various services: Authentication, Firestore, Realtime Database, Storage, Hosting, Functions, and Machine Learning. The main content area features a blue background with the text 'Auth Flutter' and 'Get started by adding Firebase to your app'. It includes icons for iOS, Android, web, and Windows development, along with a button to 'Add an app to get started'. A cartoon illustration of two people interacting with a large smartphone screen is also present.

dan ini adalah tampilan dashboard project firebase

This screenshot shows the same Project Overview page as above, but the 'Build' section of the sidebar is expanded. The expanded menu lists all the services available: Authentication, Firestore, Realtime Database, Storage, Hosting, Functions, and Machine Learning. Each service has its corresponding icon next to it.

dan ini merupakan macam macam service yang di tawarkan oleh firebase

1. Authentication

Authentication mencakup proses authentication mulai dari masuk menggunakan email dan password, google, facebook dapat di handle oleh servis ini

2. Firestore

firestore mencakup database sekaligus storage atau gabungan dari realtime database dan storage

3. Realtime Database

realtime database adalah database yang memiliki bentuk json dalam penyimpanannya sehingga sangat fleksibel dengan data, dan tidak terbatas dengan table dan relational nya

4. Storage

storage merupakan proses untuk menyimpan asset yang berupa data gambar, video, music dan lain lain, dapat disimpan di servis ini, dan setiap item yang di simpan memiliki url sendiri sendiri berdasarkan id yang telah di generate oleh firebase

5. Hosting

6. Hosting adalah servis yang disediakan firebase untuk kita mengupload file web kita yang berupa html

7. Functions

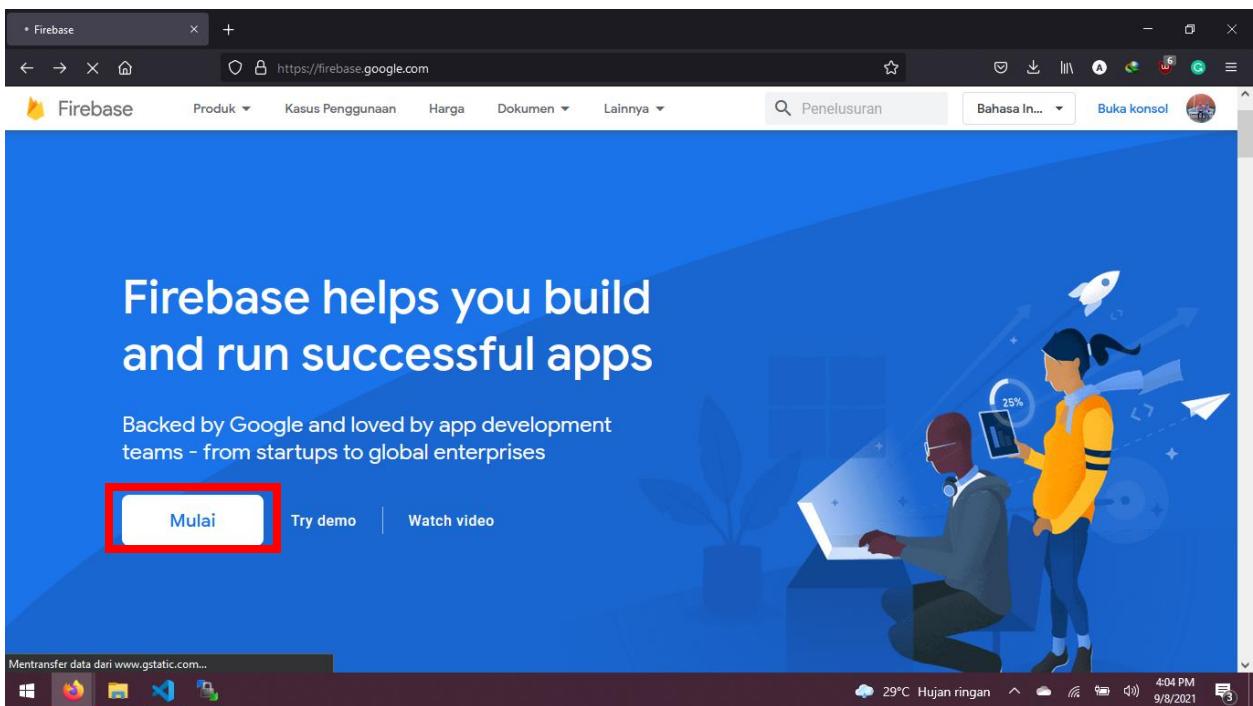
8. Machine Learning

d. Alat dan Bahan

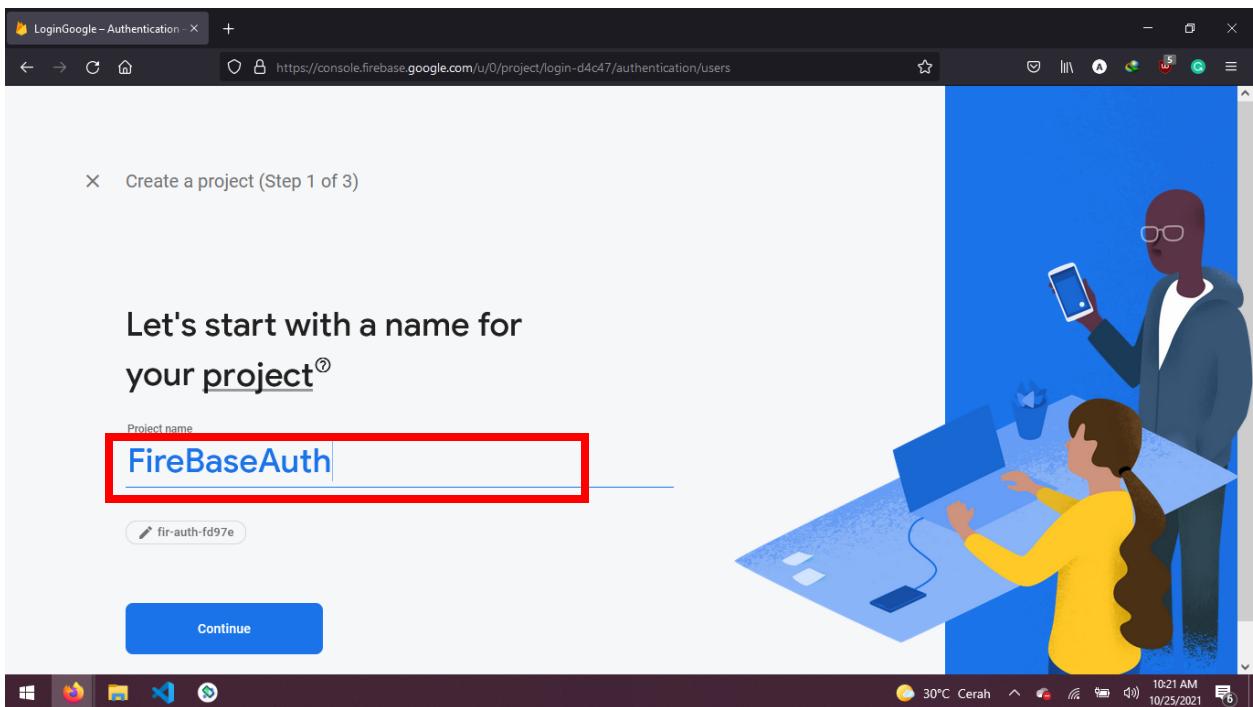
1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

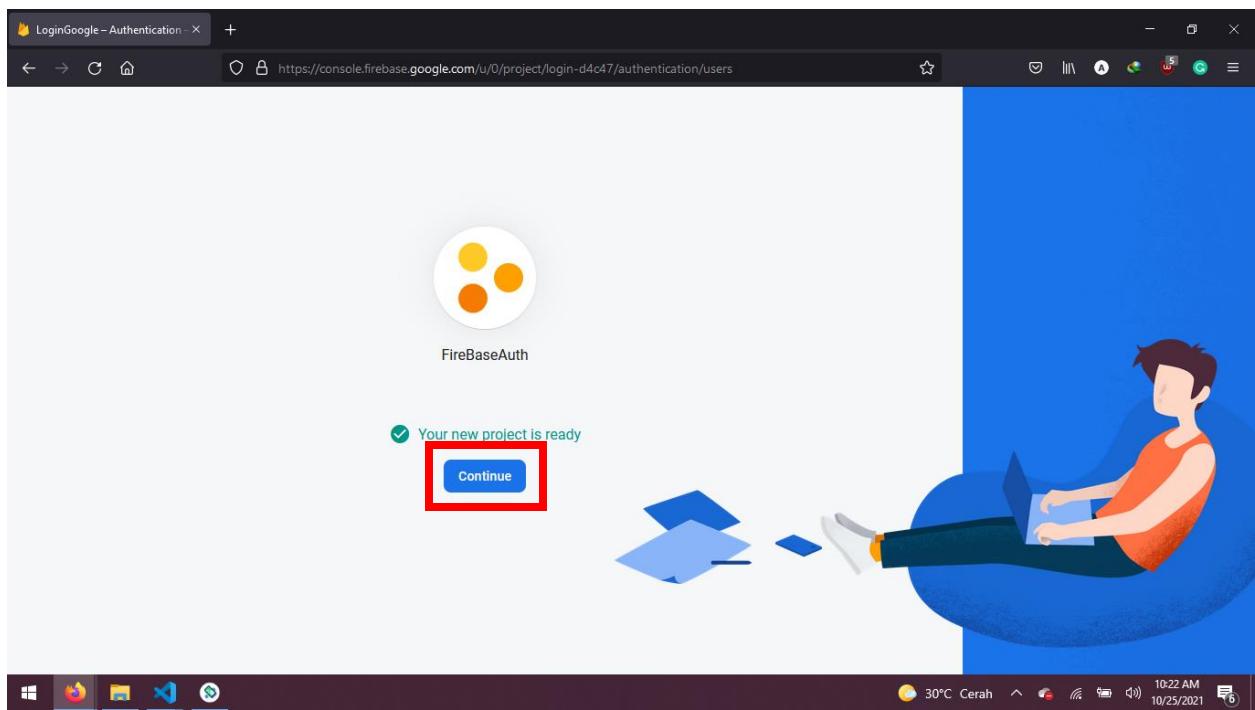
1. Membuat project flutter dengan nama minggu_11
2. Buka link berikut : <https://firebase.google.com/>



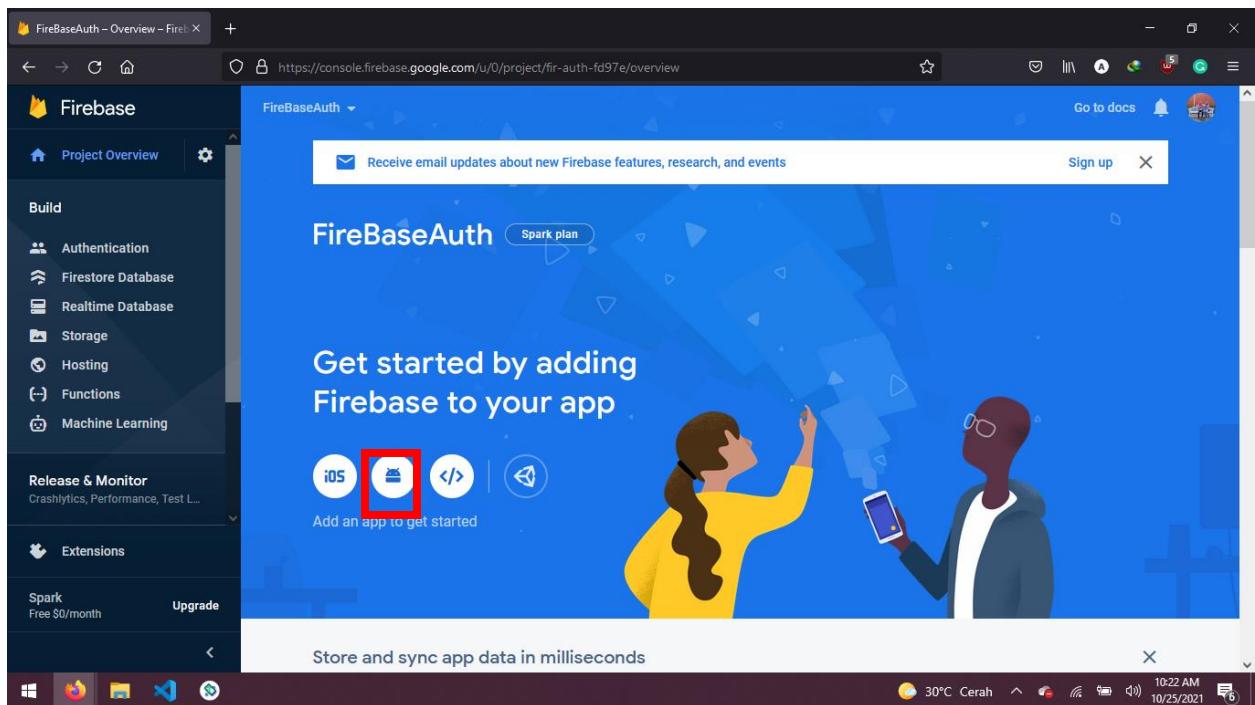
2. Membuat Project dengan nama “LoginGoogle”, kemudian tekan tombol **continue**.



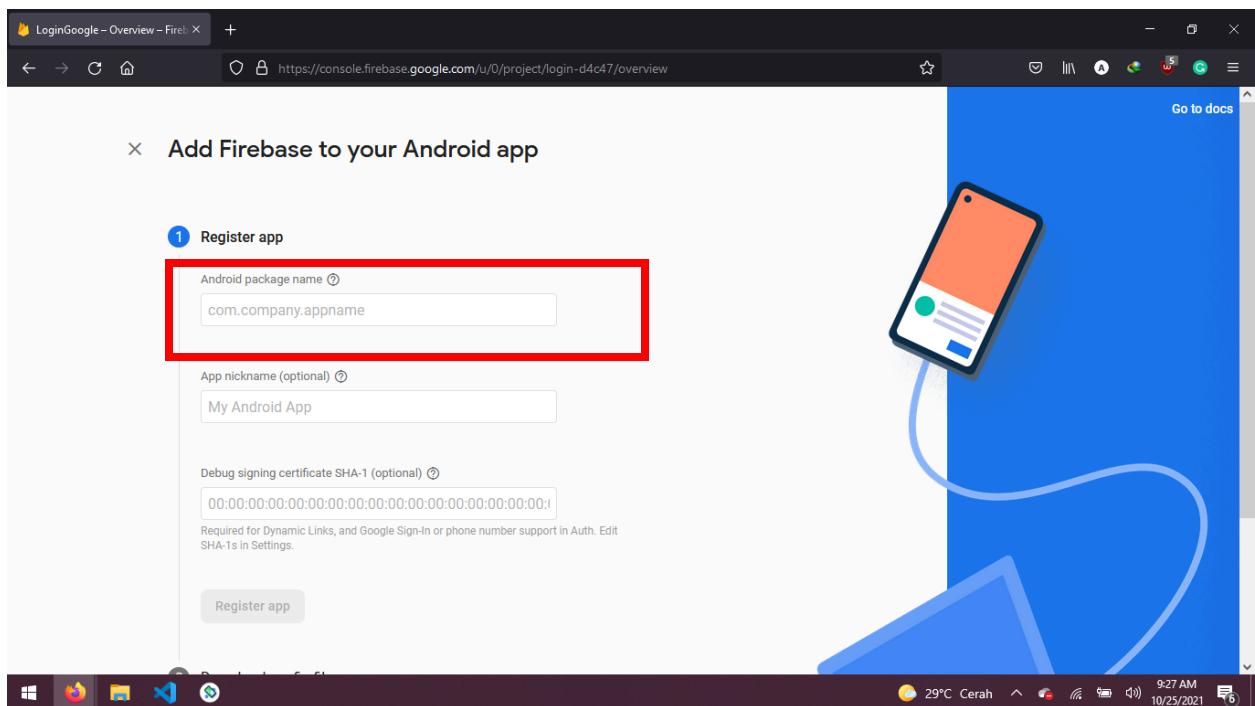
3. Tekan tombol **continue** hingga step 3.



4. Memilih platform android.



5. Memasukkan android packagenya.



The screenshot shows the Android Studio interface. On the left is the project navigation pane with 'MINGGU_11' selected, showing modules like '.dart_tool', '.idea', 'android', 'app', 'src', 'debug', and 'main'. Under 'main', files for 'java', 'kotlin', 'res', and 'AndroidManifest.xml' are listed. The 'AndroidManifest.xml' file is open in the code editor. The XML code is as follows:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android">
    <package android:label="minggu_11" android:icon="@mipmap/ic_launcher">
        <application>
            <activity android:name=".MainActivity" android:launchMode="singleTop" android:theme="@style/LaunchTheme" android:configChanges="orientation|keyboardHidden|keyboard|screenSize|smallestScreenSize|locale" android:hardwareAccelerated="true" android:windowSoftInputMode="adjustResize">
                <!-- Specifies an Android theme to apply to this Activity as soon as -->
```

package com.example.minggu_11 digunakan untuk mendaftarkan apps kita di firebase dan mendapatkan google-service.json

6. download google-services.json

2 Download config file

Instructions for Android Studio below | [Unity](#) [C++](#)

Switch to the Project view in Android Studio to see your project root directory.

Move the google-services.json file you just downloaded into your Android app module root directory.

Next

Project Structure

MyApplication (~/Desktop/MyApplication)

- build
- libs
- src
- .gitignore
- app.iml
- build.gradle
- google-services.json
- proguard-rules.pro
- gradle

kemudian pindahkan file ke dalam **C:\Users\ARVITA\dev\project\minggu_11\android\app** sesuaikan **C:\Users\ARVITA\dev\project** dengan lokasi project kalian masing-masing

| ARVITA > dev > project > minggu_11 > android > app > | | | |
|--|--------------------|-------------|------|
| Name | Date modified | Type | Size |
| src | 2/17/2022 8:40 AM | File folder | |
| build.gradle | 2/17/2022 8:40 AM | GRADLE File | 2 KB |
| google-services.json | 11/5/2021 11:30 AM | JSON File | 1 KB |

7. Tekan tombol next

3 Add Firebase SDK

Instructions for Gradle | [Unity](#) [C++](#)

The Google services plugin for [Gradle](#) loads the google-services.json file you just downloaded. Modify your build.gradle files to use the plugin.

Project-level build.gradle (<project>/build.gradle):

```
buildscript {  
    repositories {  
        // Check that you have the following line (if not, add it):  
        google() // Google's Maven repository  
    }  
    dependencies {  
        ...  
        // Add this line  
        classpath 'com.google.gms:google-services:4.3.10'  
    }  
}
```

```
allprojects {
    ...
    repositories {
        // Check that you have the following line (if not, add it):
        google() // Google's Maven repository
        ...
    }
}
```

App-level build.gradle (<project>/<app-module>/build.gradle):

App-level build.gradle (<project>/<app-module>/build.gradle):

```
apply plugin: 'com.android.application'
// Add this line
apply plugin: 'com.google.gms.google-services'

dependencies {
    // Import the Firebase BoM
    implementation platform('com.google.firebase:firebase-bom:28.4.2')

    // Add the dependencies for the desired Firebase products
    // https://firebase.google.com/docs/android/setup#available-libraries
}
```

By using the Firebase Android BoM, your app will always use compatible Firebase library versions. [Learn more](#)

Finally, press "Sync now" in the bar that appears in the IDE:

Gradle files have changed since last sync [Sync now](#)

Previous

Next

8. Tekan tombol continue to console

4 Next steps

You're all set!

Make sure to check out the [documentation](#) to learn how to get started with each Firebase product that you want to use in your app.

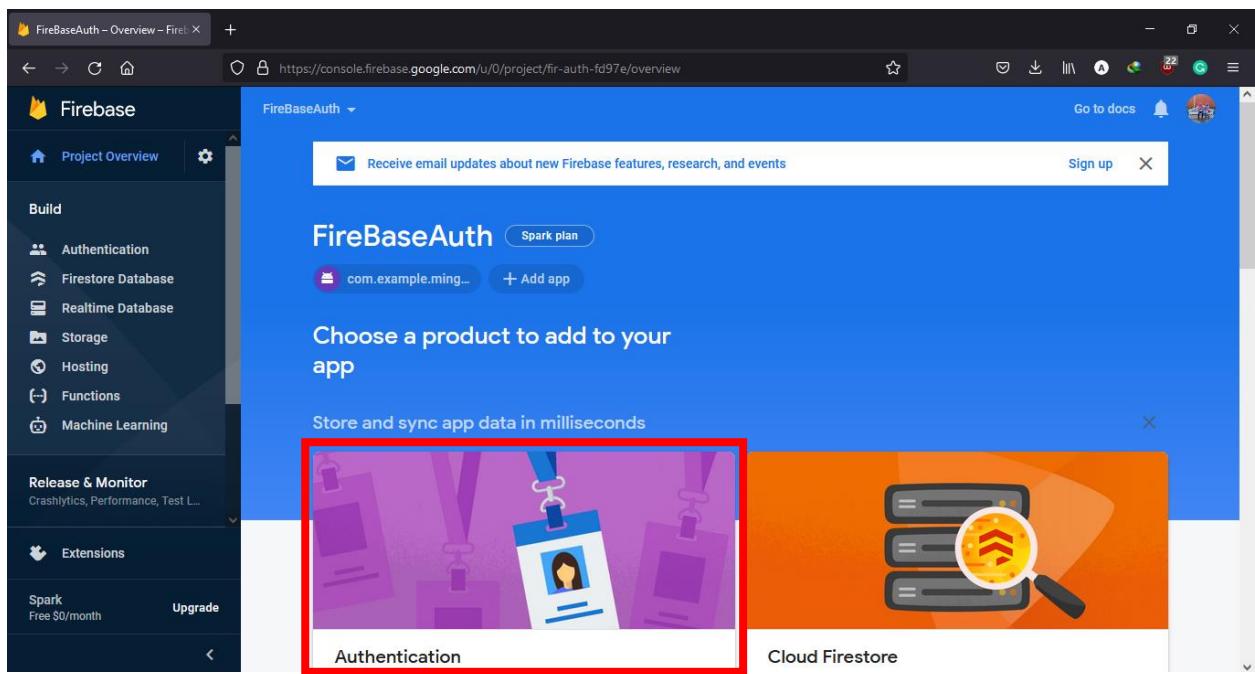
You can also explore [sample Firebase apps](#).

Or, continue to the console to explore Firebase.

Previous

Continue to console

9. Memilih product Authentication



10. Pilih sign-in methode kemudian enable Email/Password dan kemudian Save.

A screenshot of the 'Authentication' settings page in the Firebase console. The 'Sign-in method' tab is selected. Under 'Sign-in providers', the 'Email/Password' provider is listed with its 'Enable' switch turned on. Below it, the 'Email link (passwordless sign-in)' provider has its switch turned off. At the bottom right, there are 'Cancel' and 'Save' buttons, with 'Save' being highlighted with a red box.

11. Tambahkan User dan password

The screenshot shows the Firebase Authentication interface under the 'Users' tab. A search bar at the top allows searching by email address, phone number, or user UID. Below the search bar is a table with columns: Identifier, Providers, Created, Signed In, and User UID. One user is listed: aqwuarvitha@gmail.com, with a dropdown arrow next to the provider column, created on Oct 25, 2021, and a user ID k3SYXI9jkgVldGOuWMO2DZlxD5j2. At the bottom of the table, there are pagination controls for 'Rows per page: 50' and '1 - 1 of 1'.

f. Hasil dan Pembahasan

1. Dokumentasi tugas berupa laporan secara individu penjelasan pada prosedur kerja kemudian dikumpulkan dalam bentuk laporan PDF dan dikumpulkan di LMS.
2. Pengumpulan source code melalui link commit gitlab/github pada LMS.

g. Kesimpulan

Mahasiswa mendapatkan memahami Authentication, serta implementasi code Authentication.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|--------------------|--|-------|-----|-----|-----|
| 1 | Ketepatan waktu dan ketepatan implementasi pada konsep Authentication | 1 | 2 | 3 | (4) |
| 2 | Ketepatan waktu dan ketepatan implementasi pada konsep Authentication | 1 | 2 | (3) | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat implementasi pada konsep Authentication | 1 | (2) | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan implementasi pada konsep Authentication | (1) | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 38 : AUTHENTICATION

| | |
|----------------------------------|---|
| Pokok Bahasan | : Pengambilan API |
| Acara Praktikum/Pertemuan | : Minggu 11 / 2 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami konsep Authentication.
2. Mahasiswa mampu menerapkan Authentication

b. Indikator

Keberhasilan mahasiswa memahami konsep Authentication dan menerapkan Authentication pada flutter.

c. Dasar Teori

-

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

1. minggu_11->android->build.gradle

```
8     dependencies {  
9         classpath 'com.android.tools.build:gradle:4.1.0'  
10        classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"  
11        classpath 'com.google.gms:google-services:4.3.10'  
12    }  
13 }
```

2. Minggu_11->android->app->build.gradle

```
54 flutter {  
55     source '.../...'  
56 }  
57  
58 dependencies {  
59     implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"  
60     implementation platform('com.google.firebaseio:firebase-bom:28.4.2')  
61 }  
62
```

3. Minggu_11->pubspec.yaml

```
environment:  
|   sdk: ">=2.12.0 <3.0.0"  
  
dependencies:  
|   cupertino_icons: ^1.0.2  
|   firebase_auth: ^3.1.2  
|   firebase_core: ^1.7.0  
|   flutter:  
|       sdk: flutter  
|       http: ^0.13.3  
  
dev_dependencies:  
|   flutter_test:  
|       sdk: flutter
```

4. Merubah main.dart seperti berikut :

```
7 Future<void> main() async {  
8     WidgetsFlutterBinding.ensureInitialized();  
9     await Firebase.initializeApp();  
10    runApp(MyApp());  
11 }  
12  
13 class MyApp extends StatelessWidget {  
14     // This widget is the root of your application.  
15     @override  
16     Widget build(BuildContext context) {  
17         return MaterialApp(  
18             title: 'Flutter Demo',  
19             theme: ThemeData(  
20                 primarySwatch: Colors.blue,  
21             ), // ThemeData  
22             // home: LoginScreen(),  
23             home: LoginScreen(),  
24         ); // MaterialApp  
25     }  
26 }
```

f. Hasil dan Pembahasan

1. Dokumentasi tugas berupa laporan secara individu penjelasan pada prosedur kerja kemudian dikumpulkan dalam bentuk laporan PDF dan dikumpulkan di LMS.
2. Pengumpulan source code melalui link commit gitlab/github pada LMS

g. Kesimpulan

Mahasiswa mendapatkan memahami Authentication, serta implementasi code Authentication.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|--------------------|--|-------|-----|-----|-----|
| 1 | Ketepatan waktu dan ketepatan implementasi pada konsep Authentication | 1 | 2 | 3 | (4) |
| 2 | Ketepatan waktu dan ketepatan implementasi pada konsep Authentication | 1 | 2 | (3) | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat implementasi pada konsep Authentication | 1 | (2) | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan implementasi pada konsep Authentication | (1) | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 39 : AUTHENTICATION

| | |
|----------------------------------|---|
| Pokok Bahasan | : Authentication |
| Acara Praktikum/Pertemuan | : Minggu 11 / 3 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami konsep Authentication.
2. Mahasiswa mampu menerapkan Authentication

b. Indikator

Keberhasilan mahasiswa memahami konsep Authentication dan menerapkan Authentication pada flutter.

c. Dasar Teori

-

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

1. Menambahkan file dart baru yaitu LoginScreen.dart seperti berikut :

```
6  class LoginScreen extends StatefulWidget {
7    @override
8    _LoginScreenState createState() => _LoginScreenState();
9  }
10
11 class _LoginScreenState extends State<LoginScreen> {
12   final TextEditingController _emailController = TextEditingController();
13   final TextEditingController _passwordController = TextEditingController();
14
15   final FirebaseAuth _firebaseAuth = FirebaseAuth.instance;
```

```
16  registerSubmit() async {
17      try {
18          await _firebaseAuth.createUserWithEmailAndPassword(
19              email: _emailController.text.toString().trim(),
20              password: _passwordController.text);
21      } catch (e) {
22          print(e);
23          SnackBar(content: Text(e.toString()));
24      }
25  }
26
27  loginSubmit() async {
28      try {
29          _firebaseAuth
30              .signInWithEmailAndPassword(
31                  email: _emailController.text, password: _passwordController.text)
32              .then((value) => Navigator.of(context).pushReplacement(
33                  MaterialPageRoute(builder: (context) => HomeScreen())));
34      } catch (e) {
35          print(e);
36          SnackBar(content: Text(e.toString()));
37      }
38  }
39
40  @override
41  Widget build(BuildContext context) {
42      return Scaffold(
43          body: Padding(
44              padding: EdgeInsets.all(10),
45              child: ListView(
46                  children: <Widget>[
47                      Container(
48                          alignment: Alignment.center,
49                          padding: EdgeInsets.all(10),
50                          margin: const EdgeInsets.only(top: 40),
51                          child: Text(
52                              "Latihan Auth",
53                              style: TextStyle(
54                                  color: Colors.blue,
55                                  fontWeight: FontWeight.w500,
56                                  fontSize: 30), // TextStyle
57                          ), // Text
58                      ), // Container
59                      Padding(
60                          padding: const EdgeInsets.all(8),
61                          child: Image.asset(
62                              "assets/img/flutter.png",
63                              height: 100,
64                              width: 100,
```

```

65     ), // Image.asset
66     ], // Padding
67     Container(
68       padding: EdgeInsets.all(10),
69       child: TextField(
70         controller: _emailController,
71         decoration: InputDecoration(
72           border: OutlineInputBorder(), labelText: "Username"), //
73         ), // TextField // Container
74     Container(
75       padding: EdgeInsets.all(10),
76       child: TextField(
77         controller: _passwordController,
78         obscureText: true,
79         decoration: InputDecoration(
80           border: OutlineInputBorder(), labelText: "Password"), //
81         ), // TextField // Container
82     TextButton(onPressed: () {}, child: Text("Forgot password")),
83     Container(
84       height: 50,
85       padding: EdgeInsets.fromLTRB(10, 0, 10, 0),
86       child: ElevatedButton(
87         style: raisedButtonStyle,
88         child: Text("Register"),
89         onPressed: () {
90           registerSubmit();
91         },
92       ), // ElevatedButton
93     ), // Container
94     SizedBox(height: 10),
95     Container(
96       height: 50,
97       padding: EdgeInsets.fromLTRB(10, 0, 10, 0),
98       child: ElevatedButton(
99         style: raisedButtonStyle,
100        child: Text("Login"),
101        onPressed: () {
102          loginSubmit();
103        },
104      ), // ElevatedButton
105    ), // Container
106  ], // <Widget>[]
107 ), // ListView
108 ), // Padding
109 ); // Scaffold
110 }
111 }

```

```

113   final ButtonStyle raisedButtonStyle = ElevatedButton.styleFrom(
114     onPrimary: Colors.grey[300],
115     primary: Colors.blue[300],
116     minimumSize: Size(88, 36),
117     padding: EdgeInsets.symmetric(horizontal: 16),
118     shape: const RoundedRectangleBorder(
119       borderRadius: BorderRadius.all(Radius.circular(6)),
120     ); // RoundedRectangleBorder
121

```

f. Hasil dan Pembahasan

1. Dokumentasi tugas berupa laporan secara individu penjelasan pada prosedur kerja kemudian dikumpulkan dalam bentuk laporan PDF dan dikumpulkan di LMS.
2. Pengumpulan source code melalui link commit gitlab/github pada LMS

g. Kesimpulan

Mahasiswa mendapatkan memahami Authentication, serta implementasi code Authentication.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|--------------------|--|-------|-----|-----|-----|
| 1 | Ketepatan waktu dan ketepatan implementasi pada konsep Authentication | 1 | 2 | 3 | (4) |
| 2 | Ketepatan waktu dan ketepatan implementasi pada konsep Authentication | 1 | 2 | (3) | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat implementasi pada konsep Authentication | 1 | (2) | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan implementasi pada konsep Authentication | (1) | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 40 : AUTHENTICATION

| | |
|----------------------------------|--|
| Pokok Bahasan | : Pengujian JSON API MovieDB melalui Postman |
| Acara Praktikum/Pertemuan | : Minggu 11 / 4 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami konsep Authentication.
2. Mahasiswa mampu menerapkan Authentication

b. Indikator

Keberhasilan mahasiswa memahami konsep Authentication dan menerapkan Authentication pada flutter.

c. Dasar Teori

-

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

Menambahkan file dart baru yaitu HomeScreen.dart seperti berikut :

```
5
6  class HomeScreen extends StatelessWidget {
7    Future<void> _signOut() async {
8      await FirebaseAuth.instance.signOut();
9    }
10
11   @override
12   Widget build(BuildContext context) {
13     FirebaseAuth auth = FirebaseAuth.instance;
14     if (auth.currentUser != null) {
15       print(auth.currentUser!.email);
16     }
17     return Scaffold(
18       body: Padding(
```

```

19     padding: const EdgeInsets.all(30.0),
20     child: Column(
21       crossAxisAlignment: CrossAxisAlignment.start,
22       children: [
23         SizedBox(height: 60),
24         Row(
25           mainAxisAlignment: MainAxisAlignment.end,
26           children: [
27             IconButton(icon: Icon(Icons.notifications), onPressed: () {}),
28             IconButton(icon: Icon(Icons.extension), onPressed: () {})
29           ],
30         ), // Row
31         SizedBox(height: 37),
32         Text.rich(
33           TextSpan(
34             children: <TextSpan>[
35               TextSpan(
36                 text: "Welcome, \n",
37                 style: TextStyle(color: Colors.blue[300]),
38               ), // TextSpan
39               TextSpan(
40                 text: auth.currentUser!.email,
41                 style: TextStyle(color: Colors.blue[900]),
42               ), // TextSpan
43             ], // <TextSpan>[]
44           ), // TextSpan
45           style: TextStyle(fontSize: 30),
46         ), // Text.rich
47         SizedBox(height: 20),
48         TextField(
49           decoration: InputDecoration(
50             prefixIcon: Icon(Icons.search, size: 18),
51             border: OutlineInputBorder(
52               borderRadius: BorderRadius.circular(10)), // OutlineInputBorder
53             hintText: "Search"), // InputDecoration
54         ), // TextField
55         SizedBox(height: 10),
56         Text(
57           "Recomended Place",
58           style: TextStyle(fontWeight: FontWeight.w600, fontSize: 20),
59         ), // Text
60         Container(
61           child: ElevatedButton(
62             onPressed: () {
63               signOut().then((value) => Navigator.of(context)
64                 .pushReplacement(MaterialPageRoute(
65                   builder: (context) => LoginScreen())));
66             },
67             child: Text('Logout'),
68           ), // ElevatedButton // Container
69         ],
70       ), // Column
71     ), // Padding
72   ); // Scaffold
73 }
74 }
```

```

75
76     final countries = ["Tokyo", "Berlin", "Roma", "Monas"];
77

```

f. Hasil dan Pembahasan

1. Dokumentasi tugas berupa laporan secara individu penjelasan pada prosedur kerja kemudian dikumpulkan dalam bentuk laporan PDF dan dikumpulkan di LMS.
2. Pengumpulan source code melalui link commit gitlab/github pada LMS

g. Kesimpulan

Mahasiswa mendapatkan memahami Authentication, serta implementasi code Authentication.

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|--------------------|--|-------|-----|-----|-----|
| 1 | Ketepatan waktu dan ketepatan implementasi pada konsep Authentication | 1 | 2 | 3 | (4) |
| 2 | Ketepatan waktu dan ketepatan implementasi pada konsep Authentication | 1 | 2 | (3) | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat implementasi pada konsep Authentication | 1 | (2) | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan implementasi pada konsep Authentication | (1) | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 41 : CRUD

| | |
|----------------------------------|---|
| Pokok Bahasan | : Create, Read, Update dan Delete (CRUD) |
| Acara Praktikum/Pertemuan | : Minggu 12 / 1 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

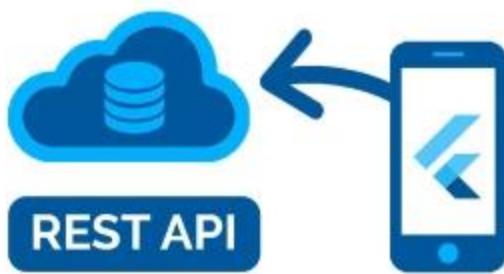
a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami Create, Read, Update dan Delete (CRUD).
2. Mahasiswa mampu menerapkan Create, Read, Update dan Delete (CRUD).

b. Indikator

Keberhasilan mahasiswa memahami serta menerapkan konsep Create, Read, Update dan Delete (CRUD), dengan implementasi source code pada kasus tertentu.

c. Dasar Teori

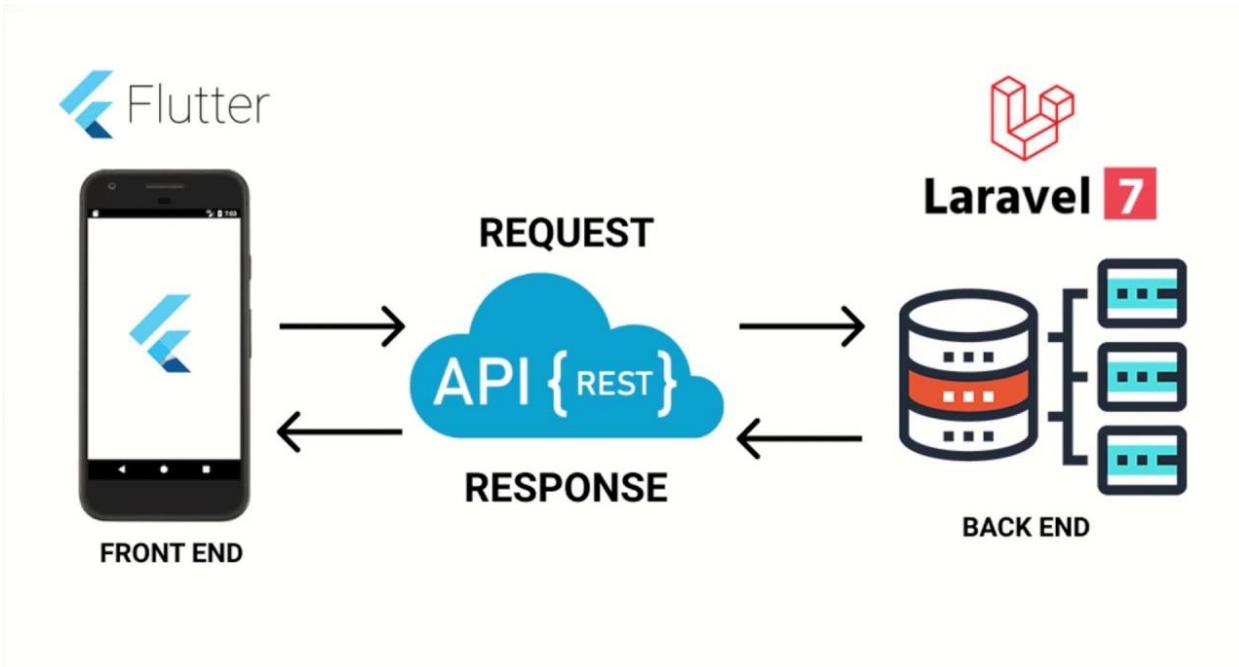


API adalah singkatan dari *Application Programming Interface*, dan memungkinkan developer untuk mengintegrasikan dua bagian dari aplikasi atau dengan aplikasi yang berbeda secara bersamaan.

Dalam sebuah aplikasi proses yang terjadi biasanya ada Create, Read, Update dan Delete atau biasa kita kenal dengan CRUD. didalam API terdapat proses :

- POST untuk menginput data (Create)
- GET untuk menampilkan data (Read)
- PUT untuk mengubah data (Update)

- DELETE untuk menghapus data (Delete)



d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

- **Membuat API**

1. Kerjakan yang ada di BKPM WEB Framework Minggu ke 12
2. Tambahkan syntax pada file VerifyCsrfToken.php pada folder App/Middleware seperti berikut ini :

```

3 namespace App\Http\Middleware;
4
5 use Illuminate\Foundation\Http\Middleware\VerifyCsrfToken as Middleware;
6
7 You, seconds ago | 1 author (You)
8 class VerifyCsrfToken extends Middleware
9 {
10     /**
11      * Indicates whether the XSRF-TOKEN cookie should be set on the response.
12      *
13      * @var bool
14      */
15     protected $addHttpCookie = true;
16
17     /**
18      * The URIs that should be excluded from CSRF verification.
19      *
20      * @var array
21      */
22     protected $except = [
23         '*',
24     ];

```

f. Hasil dan Pembahasan

1. Dokumentasi tugas berupa laporan secara individu penjelasan pada prosedur kerja kemudian dikumpulkan dalam bentuk laporan PDF dan dikumpulkan di LMS.
2. Pengumpulan source code melalui link commit gitlab/github pada LMS.

g. Kesimpulan

Mahasiswa mendapatkan memahami Create, Read, Update dan Delete (CRUD), serta implementasi code Create, Read, Update dan Delete (CRUD).

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|----|--|-------|---|---|---|
| 1 | Ketepatan waktu dan ketepatan dalam implementasi code serta menjelaskan secara detail setiap baris code Create, Read, Update dan Delete (CRUD). | 1 | 2 | 3 | 4 |
| 2 | Ketepatan waktu dan ketepatan dalam implementasi code serta menjelaskan secara singkat setiap baris code Create, Read, Update dan Delete (CRUD). | 1 | 2 | 3 | 4 |

| | | | | | |
|--------------------|--|-----|-----|---|---|
| 3 | Ketepatan waktu akan tetapi kurang tepat dalam implementasi code serta menjelaskan setiap baris code Create, Read, Update dan Delete (CRUD). | 1 | (2) | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan dalam implementasi code serta menjelaskan setiap baris code Create, Read, Update dan Delete (CRUD). | (1) | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 42 : CRUD

| | |
|----------------------------------|---|
| Pokok Bahasan | : Create, Read, Update dan Delete (CRUD) |
| Acara Praktikum/Pertemuan | : Minggu 12 / 2 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami Create, Read, Update dan Delete (CRUD).
2. Mahasiswa mampu menerapkan Create, Read, Update dan Delete (CRUD).

b. Indikator

Keberhasilan mahasiswa memahami serta menerapkan konsep Create, Read, Update dan Delete (CRUD), dengan implementasi source code pada kasus tertentu.

c. Dasar Teori

-

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

- **Setting URL Server**

1. Pertama buka command prompt(cmd) , ketikkan “ipconfig” sehingga muncul ip address seperti berikut :

```
Command Prompt
Windows IP Configuration

Wireless LAN adapter Local Area Connection* 1:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 2:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:
  Connection-specific DNS Suffix  . :
  IPv6 Address . . . . . : 2001:448a:5122:af76:94be:c14f:759:b529
  Temporary IPv6 Address . . . . . : 2001:448a:5122:af76:954b:b50c:5a2:9454
  Link-local IPv6 Address . . . . . : fe80::94be:c14f:759:b529%9
  IPv4 Address. . . . . : 192.168.1.15
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . : fe80::1%9
                           192.168.1.1

Ethernet adapter Bluetooth Network Connection:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix  . :

C:\Users\ARVITA>_
```

2. Kemudian masuk ke folder project laravel masing-masing

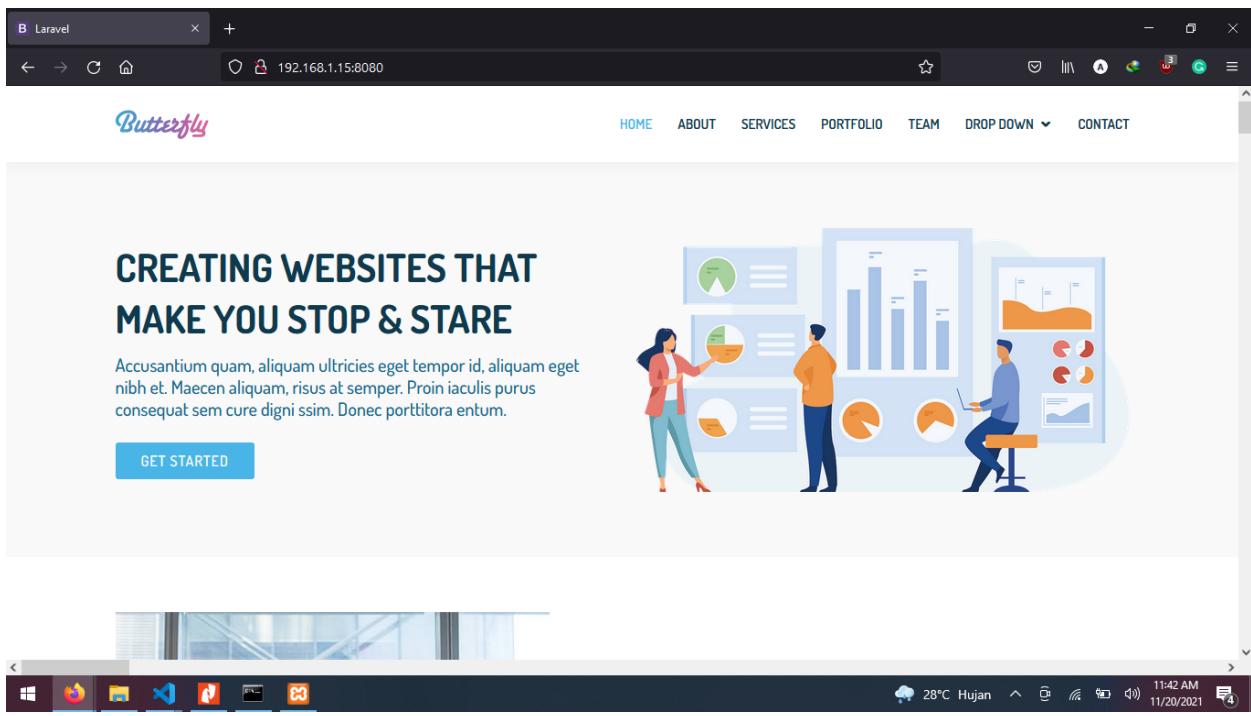
```
C:\Users\ARVITA>cd C:\xampp\htdocs\BelajarLaravel

C:\xampp\htdocs\BelajarLaravel>_
```

3. Jalankan syntax php artisan serve sesuai dengan ip address masing-masing seperti berikut:

```
C:\xampp\htdocs\BelajarLaravel>php artisan serve --host 192.168.1.15 --port 8080
Laravel development server started: http://192.168.1.15:8080
[Sat Nov 20 11:40:49 2021] PHP 7.4.15 Development Server (http://192.168.1.15:8080) started
-
```

sehingga apabila dijalankan maka akan muncul tampilan laravel seperti berikut ini :



f. Hasil dan Pembahasan

1. Dokumentasi tugas berupa laporan secara individu penjelasan pada prosedur kerja kemudian dikumpulkan dalam bentuk laporan PDF dan dikumpulkan di LMS.
2. Pengumpulan source code melalui link commit gitlab/github pada LMS

g. Kesimpulan

Mahasiswa mendapatkan memahami Create, Read, Update dan Delete (CRUD), serta implementasi code Create, Read, Update dan Delete (CRUD).

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|----|--|-------|---|---|---|
| 1 | Ketepatan waktu dan ketepatan dalam implementasi code serta menjelaskan secara detail setiap baris code Create, Read, Update dan Delete (CRUD). | 1 | 2 | 3 | 4 |
| 2 | Ketepatan waktu dan ketepatan dalam implementasi code serta menjelaskan secara singkat setiap baris code Create, Read, Update dan Delete (CRUD). | 1 | 2 | 3 | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat dalam implementasi code serta menjelaskan setiap baris code Create, Read, Update dan Delete (CRUD). | 1 | 2 | 3 | 4 |

| | | |
|--------------------|---|---|
| 4 | <p>Keterlambatan pengumpulan tugas dan ketidaktepatan dalam implementasi code serta menjelaskan setiap baris code Create, Read, Update dan Delete (CRUD).</p> | <input checked="" type="radio"/> 1 2 3 4 |
| Jumlah skor | | |

ACARA 43 : CRUD

| | |
|----------------------------------|---|
| Pokok Bahasan | : Create, Read, Update dan Delete (CRUD) |
| Acara Praktikum/Pertemuan | : Minggu 12 / 3 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami Create, Read, Update dan Delete (CRUD).
2. Mahasiswa mampu menerapkan Create, Read, Update dan Delete (CRUD).

b. Indikator

Keberhasilan mahasiswa memahami serta menerapkan konsep Create, Read, Update dan Delete (CRUD), dengan implementasi source code pada kasus tertentu.

c. Dasar Teori

-

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

1. Buat project flutter dengan nama crud
2. Tambahkan <http://0.0.0.0:8080> pada pubspec.yaml
3. Buat class baru dengan nama adddata.dart, tambahkan sintak seperti berikut:

```

lib > adddata.dart > _AddDataState > addData
1   import 'package:flutter/material.dart';
2   import 'package:http/http.dart' as http;
3
4   import 'main.dart';
5
6   class AddData extends StatefulWidget {
7     @override
8     _AddDataState createState() => new _AddDataState();
9   }
10
11  class _AddDataState extends State<AddData> {
12    TextEditingController controllerNama = new TextEditingController();
13    TextEditingController controllerTingkatan = new TextEditingController();
14    TextEditingController controllerTahunMasuk = new TextEditingController();
15    TextEditingController controllerTahunKeluar = new TextEditingController();
16
17    void addData() {
18      var url = "http://192.168.1.15:8080/api/api_pendidikan/";
19
20      http.post(Uri.parse(url), body: {
21        "nama": controllerNama.text,
22        "tingkatan": controllerTingkatan.text,
23        "tahun_masuk": controllerTahunMasuk.text,
24        "tahun keluar": controllerTahunKeluar.text
25      });
26    }
27
28    @override
29    Widget build(BuildContext context) {
30      return new Scaffold(
31        appBar: new AppBar(
32          title: new Text("ADD DATA"),
33        ), // AppBar
34        body: Padding(
35          padding: const EdgeInsets.all(10.0),
36          child: ListView(
37            children: <Widget>[
38              new Column(
39                children: <Widget>[
40                  new TextField(
41                    controller: controllerNama,
42                    decoration:
43                      new InputDecoration(hintText: "Nama", labelText: "Nama"),
44                  ), // TextField
45                  new TextField(
46                    controller: controllerTingkatan,
47                    decoration: new InputDecoration(

```

```

41     controller: controllerNama,
42     decoration:
43       | new InputDecoration(hintText: "Nama", labelText: "Nama"),
44     ), // TextField
45     new TextField(
46       controller: controllerTingkatan,
47       decoration: new InputDecoration(
48         hintText: "Tingkatan", labelText: "Tingkatan"), // InputDecoration
49     ), // TextField
50     new TextField(
51       controller: controllerTahunMasuk,
52       decoration: new InputDecoration(
53         hintText: "Tahun Masuk", labelText: "Tahun Masuk"), // InputDecoration
54     ), // TextField
55     new TextField(
56       controller: controllerTahunKeluar,
57       decoration: new InputDecoration(
58         hintText: "Tahun Keluar", labelText: "Tahun Keluar"), // InputDecoration
59     ), // TextField
60     new Padding(
61       padding: const EdgeInsets.all(10.0),
62     ), // Padding
63   ), // Padding
64   new RaisedButton(
65     child: new Text("ADD DATA"),
66     color: Colors.blueAccent,
67     onPressed: () {
68       addData();
69       Navigator.of(context).push(new MaterialPageRoute(
70         builder: (BuildContext context) => new Home()), // MaterialPageRoute
71       ),
72     ), // RaisedButton
73   ], // <Widget>[]
74 ], // <Widget>[]
75 ), // ListView
76 ), // Padding
77 ); // Scaffold
78 }
79 }
80

```

Sesuaikan url dengan ip address pada masing-masing device. Gunakan jaringan yang sama antara mobile dengan database server.

4. Buat class baru dengan nama editdata.dart, tambahkan sintak seperti berikut:

```

lib > editdata.dart > _EditDataState > editData
1 import 'dart:convert';
2
3 import 'package:flutter/material.dart';
4 import 'package:http/http.dart' as http;
5 import './main.dart';
6
7 class EditData extends StatefulWidget {
8   final List list;
9   final int index;
10
11   EditData({required this.list, required this.index, id});
12
13   @override
14   _EditDataState createState() => new _EditDataState();
15 }
16
17 class _EditDataState extends State<EditData> {
18   late TextEditingController controllerNama;
19   late TextEditingController controllerTingakatan;
20   late TextEditingController controllerTahunMasuk;
21   late TextEditingController controllerTahunKeluar;
22
23   void editData() async {
24     http.put(
25       Uri.parse('http://192.168.1.15:8080/api/api_pendidikan/' +
26         widget.list[widget.index]['id'].toString()),
27       headers: {
28         'Accept': 'application/json',
29       },
30       body: {
31         "nama": controllerNama.text,
32         "tingkatan": controllerTingakatan.text,
33         "tahun_masuk": controllerTahunMasuk.text,
34         "tahun_keluar": controllerTahunKeluar.text
35       }).then((response) {
36         print('Response status : ${response.statusCode}');
37         print('Response body : ${response.body}');
38       });
39     }
40
41   @override
42   void initState() {
43     controllerNama =
44       new TextEditingController(text: widget.list[widget.index]['nama']);

```

```

45     controllerTingakatan = new TextEditingController(
46         text: widget.list[widget.index]['tingkatan'].toString());
47     controllerTahunMasuk = new TextEditingController(
48         text: widget.list[widget.index]['tahun_masuk'].toString());
49     controllerTahunKeluar = new TextEditingController(
50         text: widget.list[widget.index]['tahun_keluar'].toString());
51     super.initState();
52 }
53
54 @override
55 Widget build(BuildContext context) {
56     return new Scaffold(
57         appBar: new AppBar(
58             title: new Text("EDIT DATA"),
59         ), // AppBar
60         body: Padding(
61             padding: const EdgeInsets.all(10.0),
62             child: ListView(
63                 children: <Widget>[
64                     new Column(
65                         children: <Widget>[
66                             new TextField(
67                                 controller: controllerNama,
68                                 decoration:
69                                     new InputDecoration(hintText: "Nama", labelText: "Nama"),
70                             ), // TextField
71                             new TextField(
72                                 controller: controllerTingakatan,
73                                 decoration: new InputDecoration(
74                                     hintText: "Tingkatan", labelText: "Tingkatan"), // InputD
75                             ), // TextField
76                             new TextField(
77                                 controller: controllerTahunMasuk,
78                                 decoration: new InputDecoration(
79                                     hintText: "Tahun Masuk", labelText: "Tahun Keluar"), // I
80                             ), // TextField
81                             new TextField(
82                                 controller: controllerTahunKeluar,
83                                 decoration: new InputDecoration(
84                                     hintText: "Tahun Keluar", labelText: "Tahun Keluar"), // I
85                             ), // TextField
86                             new Padding(
87                                 padding: const EdgeInsets.all(10.0),

```

```

87         padding: const EdgeInsets.all(10.0),
88     ), // Padding
89     new RaisedButton(
90         child: new Text("EDIT DATA"),
91         color: Colors.blueAccent,
92         onPressed: () {
93             editData();
94             Navigator.of(context).push(new MaterialPageRoute(
95                 builder: (BuildContext context) => new Home())); // Mat
96         },
97     ), // RaisedButton
98     ], // <Widget>[]
99     ), // Column
100    ], // <Widget>[]
101   ), // ListView
102   ), // Padding
103 ); // Scaffold
104 }
105 }
```

Sesuaikan url dengan ip address pada masing-masing device. Gunakan jaringan yang sama antara mobile dengan database server.

f. Hasil dan Pembahasan

1. Dokumentasi tugas berupa laporan secara individu penjelasan pada prosedur kerja kemudian dikumpulkan dalam bentuk laporan PDF dan dikumpulkan di LMS.
2. Pengumpulan source code melalui link commit gitlab/github pada LMS

g. Kesimpulan

Mahasiswa mendapatkan memahami Create, Read, Update dan Delete (CRUD), serta implementasi code Create, Read, Update dan Delete (CRUD).

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|----|--|-------|-----|-----|-----|
| 1 | Ketepatan waktu dan ketepatan dalam implementasi code serta menjelaskan secara detail setiap baris code Create, Read, Update dan Delete (CRUD). | 1 | 2 | 3 | (4) |
| 2 | Ketepatan waktu dan ketepatan dalam implementasi code serta menjelaskan secara singkat setiap baris code Create, Read, Update dan Delete (CRUD). | 1 | 2 | (3) | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat dalam implementasi code serta menjelaskan setiap baris code Create, Read, Update dan Delete (CRUD). | 1 | (2) | 3 | 4 |

| | | |
|--------------------|---|---|
| 4 | <p>Keterlambatan pengumpulan tugas dan ketidaktepatan dalam implementasi code serta menjelaskan setiap baris code Create, Read, Update dan Delete (CRUD).</p> | <input checked="" type="radio"/> 1 2 3 4 |
| Jumlah skor | | |

ACARA 44 : CRUD

| | |
|----------------------------------|---|
| Pokok Bahasan | : Create, Read, Update dan Delete (CRUD) |
| Acara Praktikum/Pertemuan | : Minggu 12 / 4 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami Create, Read, Update dan Delete (CRUD).
2. Mahasiswa mampu menerapkan Create, Read, Update dan Delete (CRUD).

b. Indikator

Keberhasilan mahasiswa memahami serta menerapkan konsep Create, Read, Update dan Delete (CRUD), dengan implementasi source code pada kasus tertentu.

c. Dasar Teori

-

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

1. Buat class baru dengan nama detail.dart, tambahkan sintak seperti berikut:

```

lib > detail.dart > _DetailState > deleteData
  1 import 'package:flutter/material.dart';
  2 import 'package:http/http.dart' as http;
  3
  4 import './editdata.dart';
  5 import './main.dart';
  6
  7 class Detail extends StatefulWidget {
  8   List list;
  9   int index;
 10  Detail({
 11    Key? key,
 12    required this.list,
 13    required this.index,
 14  }) : super(key: key);
 15  @override
 16  _DetailState createState() => new _DetailState();
 17 }
 18
 19 class _DetailState extends State<Detail> {
 20   void deleteData() {
 21     var url = 'http://192.168.1.15:8080/api/api_pendidikan/';
 22     http.delete(Uri.parse(url + widget.list[widget.index]['id'].toString()),
 23       body: {
 24         'id': widget.list[widget.index]['id'].toString()
 25       }).then((response) {
 26         print('Response status : ${response.statusCode}');
 27         print('Response body : ${response.body}');
 28     });
 29   }
 30
 31   void confirm() {
 32     AlertDialog alertDialog = new AlertDialog(
 33       content: new Text(
 34         "Are You sure want to delete '${widget.list[widget.index]['nama']}'"),
 35       actions: <Widget>[
 36         new RaisedButton(
 37           child: new Text(
 38             "OK DELETE!",
 39             style: new TextStyle(color: Colors.black),
 40           ), // Text
 41           color: Colors.red,
 42           onPressed: () {
 43             deleteData();
 44             Navigator.of(context).push(new MaterialPageRoute(

```

```

45     builder: (BuildContext context) => new Home(),
46   );
47 },
48 ),
49 new RaisedButton(
50   child: new Text("CANCEL", style: new TextStyle(color: Colors.black)),
51   color: Colors.green,
52   onPressed: () => Navigator.pop(context),
53 ),
54 ],
55 );
56 );
57 showDialog(builder: (context) => alertDialog, context: context);
58 }
59
60 @override
61 Widget build(BuildContext context) {
62   return new Scaffold(
63     appBar:
64       new AppBar(title: new Text("${widget.list[widget.index]['nama']}")),
65     body: new Container(
66       height: 270.0,
67       padding: const EdgeInsets.all(20.0),
68       child: new Card(
69         child: new Center(
70           child: new Column(
71             children: <Widget>[
72               new Padding(
73                 padding: const EdgeInsets.only(top: 30.0),
74               ),
75               new Text(
76                 widget.list[widget.index]['nama'],
77                 style: new TextStyle(fontSize: 20.0),
78               ),
79               new Text(
80                 "Tingkatan : ${widget.list[widget.index]['tingkatan']}",
81                 style: new TextStyle(fontSize: 18.0),
82               ),
83               new Text(
84                 "Tahun Masuk : ${widget.list[widget.index]['tahun_masuk']}",
85                 style: new TextStyle(fontSize: 18.0),
86               ),
87               new Text(

```

```

88         "Tahun Keluar : ${widget.list[widget.index]['tahun_keluar']}",
89         style: new TextStyle(fontSize: 18.0),
90     ), // Text
91     new Padding(
92         padding: const EdgeInsets.only(top: 30.0),
93     ), // Padding
94     new Row(
95         mainAxisAlignment: MainAxisAlignment.min,
96         children: <Widget>[
97             new RaisedButton(
98                 child: new Text("EDIT"),
99                 color: Colors.green,
100                onPressed: () =>
101                    Navigator.of(context).push(new MaterialPageRoute(
102                        builder: (BuildContext context) => new EditData(
103                            list: widget.list,
104                            index: widget.index,
105                            ), // EditData
106                            )), // MaterialPageRoute
107                            ), // RaisedButton
108             new RaisedButton(
109                 child: new Text("DELETE"),
110                 color: Colors.red,
111                 onPressed: () => confirm(),
112                 ), // RaisedButton
113             ], // <Widget>[]
114         ) // Row
115     ], // <Widget>[]
116     ), // Column
117     ), // Center
118     ), // Card
119     ), // Container
120 ); // Scaffold
121 }
122 }
123

```

Sesuaikan url dengan ip address pada masing-masing device. Gunakan jaringan yang sama antara mobile dengan database server.

- Buat class baru dengan nama main.dart, tambahkan sintak seperti berikut:

```

lib > main.dart > _HomeState > getData
  1 import 'dart:async';
  2 import 'dart:convert';
  3
  4 import 'package:flutter/material.dart';
  5 import 'package:http/http.dart' as http;
  6 import './Detail.dart';
  7 import './adddata.dart';
  8
  9   Run | Debug | Profile
10 void main() {
11   runApp(new MaterialApp(
12     title: "My Store",
13     home: new Home(),
14   )); // MaterialApp
15 }
16
17 class Home extends StatefulWidget {
18   @override
19   _HomeState createState() => new _HomeState();
20 }
21
22 class _HomeState extends State<Home> {
23   Future<List> getData() async {
24     final response = await http
25       .get(Uri.parse('http://192.168.1.15:8080/api/api_pendidikan'));
26     return json.decode(response.body);
27   }
28
29   @override
30   Widget build(BuildContext context) {
31     return new Scaffold(
32       appBar: new AppBar(
33         title: new Text("Pendidikan"),
34       ), // AppBar
35       floatingActionButton: new FloatingActionButton(
36         child: new Icon(Icons.add),
37         onPressed: () => Navigator.of(context).push(new MaterialPageRoute(
38           builder: (BuildContext context) => new AddData(),
39         )), // MaterialPageRoute
40       ), // FloatingActionButton
41       body: new FutureBuilder<List>(
42         future: getData(),
43         builder: (context, snapshot) {

```

```

45     |         return snapshot.hasData
46     |         ? new ItemList(
47     |             list: snapshot.requireData,
48     |         ) // ItemList
49     |         : new Center(
50     |             child: new CircularProgressIndicator(),
51     |         ); // Center
52     |     ],
53     | ), // FutureBuilder
54   ); // Scaffold
55 }
56 }
57
58 class ItemList extends StatelessWidget {
59   final List list;
60   ItemList({required this.list});
61
62   @override
63   Widget build(BuildContext context) {
64     return new ListView.builder(
65       itemCount: list == null ? 0 : list.length,
66       itemBuilder: (context, i) {
67         return new Container(
68           padding: const EdgeInsets.all(10.0),
69           child: new GestureDetector(
70             onTap: () => Navigator.of(context).push(new MaterialPageRoute(
71               builder: (BuildContext context) => new Detail(
72                 list: list,
73                 index: i,
74               )), // Detail // MaterialPageRoute
75             child: new Card(
76               child: new ListTile(
77                 title: new Text(list[i]['nama']),
78                 leading: new Icon(Icons.widgets),
79                 subtitle: new Text("Tingkatan : ${list[i]['tingkatan']}"),
80               ), // ListTile
81             ), // Card
82           ), // GestureDetector
83         ); // Container
84     ],
85   ); // ListView.builder
86 }
87 }
88

```

Sesuaikan url dengan ip address pada masing-masing device. Gunakan jaringan yang sama antara mobile dengan database server

f. Hasil dan Pembahasan

1. Dokumentasi tugas berupa laporan secara individu penjelasan pada prosedur kerja kemudian dikumpulkan dalam bentuk laporan PDF dan dikumpulkan di LMS.
2. Pengumpulan source code melalui link commit gitlab/github pada LMS.

g. Kesimpulan

Mahasiswa mendapatkan memahami Create, Read, Update dan Delete (CRUD), serta implementasi code Create, Read, Update dan Delete (CRUD)..

h. Rubrik Penilaian

| No | Indikator | Skor* | | | |
|--------------------|--|-------|-----|-----|-----|
| 1 | Ketepatan waktu dan ketepatan dalam implementasi code serta menjelaskan secara detail setiap baris code Create, Read, Update dan Delete (CRUD). | 1 | 2 | 3 | (4) |
| 2 | Ketepatan waktu dan ketepatan dalam implementasi code serta menjelaskan secara singkat setiap baris code Create, Read, Update dan Delete (CRUD). | 1 | 2 | (3) | 4 |
| 3 | Ketepatan waktu akan tetapi kurang tepat dalam implementasi code serta menjelaskan setiap baris code Create, Read, Update dan Delete (CRUD). | 1 | (2) | 3 | 4 |
| 4 | Keterlambatan pengumpulan tugas dan ketidaktepatan dalam implementasi code serta menjelaskan setiap baris code Create, Read, Update dan Delete (CRUD). | (1) | 2 | 3 | 4 |
| Jumlah skor | | | | | |

ACARA 45 : BUILD AND RELEASE APLIKASI + PROJECT FINAL

| | |
|----------------------------------|---|
| Pokok Bahasan | : Unggah Aplikasi ke Play Store |
| Acara Praktikum/Pertemuan | : Minggu 13 / 1 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

Mahasiswa mampu mengunggah aplikasi ke play store.

b. Indikator

Keberhasilan mahasiswa mengunggah aplikasi ke play store.

c. Dasar Teori

Persiapan Unggah

Sebelum mulai mengunggah ke google play developer console, pastikan telah membaca dokumentasi dari flutter mengenai build and release an Android App pada link berikut: [Build and release an Android app | Flutter](#)

Unggah ke play store developer console

Setelah melakukan tahap-tahap dalam registrasi di atas sekarang Anda dapat mencoba untuk mengupload atau merilis aplikasi Android pada Google Play Store. Ini tidak memerlukan waktu yang cukup lama hanya tinggal menunggu beberapa waktu. Ikuti langkah-langkah berikut untuk merilis aplikasi android Anda sendiri dengan resmi :

- Login atau masuk.

Klik pada tombol “Publish an Android App on Google Play” di halaman utama Google Play Developer Console.

- Memilih opsi/pilihan

Pilih “Default language” yang diinginkan beserta judul aplikasinya. Jika aplikasinya berbahasa Indonesia, pilih saja bahasa Indonesia untuk “Default language”.

- Upload

Selanjutnya, klik tombol “Upload APK” dan lalu unggah file APK dari aplikasi kamu. Siapkan juga file keystore yang Anda gunakan untuk membuild file apk tersebut dan simpan jangan sampai hilang karena file tersebut akan dibutuhkan kembali ketika Anda ingin mengupdate atau melakukan perubahan pada aplikasi Anda. Jika file keystore yang digunakan berbeda maka aplikasi di google dev tidak akan dapat diperbarui dan menolak update aplikasi Anda. Pengertian dari file keystore adalah file yang digunakan ketika menggenerate file APK.

- Persiapkan Gambar Untuk Aplikasi

Terdapat tiga langkah untuk mempersiapkan gambar untuk aplikasi Anda. Gambar ini tidak boleh sembarangan karena harus memenuhi beberapa kriteria. Berikut kriteria dari gambar tersebut :

1. Screenshot dari tampilan aplikasi Anda, maksimal 8 buah gambar. Siapkan gambar screenshot yang menarik agar aplikasi anda mendapatkan perhatian khusus dari pengguna dan mereka terpancing untuk mendownload aplikasi anda.
2. Gambar untuk icon aplikasi yang tampil di playstore. Ukuran gambar untuk icon aplikasi adalah 512px x 512px, jangan lupa untuk membuat gambar semenarik mungkin serta relevan dengan isi aplikasi anda, agar aplikasi anda banyak yang download,
3. Gambar banner. Siapkan juga gambar untuk banner aplikasi anda dengan ukuran 1024px x 512px, buat semenarik mungkin, anda bisa menggunakan aplikasi seperti photoshop, atau corel draw untuk membuatnya.

Syarat tersebut merupakan minimal persyaratan yang harus dipenuhi. Sebenarnya juga terdapat beberapa lagi persyaratan yang bersifat opsional.

- Upload Aplikasi Pada Google Play Developer

Langkah – langkah yang bisa Anda tempuh, yaitu :

1. Login ke akun Google Developer Anda.

Untuk login anda bisa mengakses halaman play.google.com/apps/publish, kemudian silahkan login menggunakan email google dan passwordnya yang anda gunakan ketika daftar.

2. Klik tombol Create Application

Silahkan pilih opsi Default language (sesuaikan dengan bahasa aplikasi android yang ingin anda upload), kemudian title, lalu isi dengan judul aplikasi Anda agar dapat dicari di Google Play Store.

- Isi data APP Release

Pada App Release Anda akan diminta untuk mengupload file apk aplikasi android Anda dan juga informasi mengenai point-point penting yang terdapat pada aplikasi Anda.

- Isi data store listing

Data yang diminta di bagian store listing adalah data yang terkait dengan tampilan (asset) dan informasi detail mengenai aplikasi android Anda.

- Isi Data Content Rating

Selanjutnya adalah mengisi data content rating, konten rating harus diisi setelah anda mengupload file apk Anda di bagian menu App Release, untuk mengisi data konten rating silahkan klik menu Content rating di bilah menu sisi kiri, kemudian tekan tombol Continue.

Untuk mengisi Data Content rating Anda akan diminta untuk memasukan email silakan masukan email dan juga konfirmasi email, selanjutnya data yang diminta pada bagian content rating jenisnya adalah multiple choice, Anda cukup memilih saja soal-soal yang diminta, silahkan ikuti dan sesuaikan dengan data aplikasi Anda sesuai dengan yang diminta.

- Mengisi Data Pricing dan Distribution

Pada bagian ini Anda akan diminta menentukan jenis aplikasi Anda (gratis/tidak), dan juga menentukan negara mana saja yang Anda ingin aplikasi anda bisa didownload. Silahkan ikuti apa yang diminta, jika sudah selesai tekan tombol Save Draft.

- Publish APK Android

Untuk melakukan publish langkahnya adalah sebagai berikut:

1. Pilih menu App Release di bilah menu sisi kiri
2. Kemudian tekan tombol Edit Release
3. Pada halaman edit release, scroll ke bawah kemudian tekan tombol Review di bagian bawah kanan, nanti tulisan tombolnya akan berganti menjadi Start Rollout to Production.Selanjutnya tekan lagi tombol Start Rollout to production, maka nanti akan muncul dialog konfirmasi dengan tulisan “Your app will now become available to all users of the Play Store. Do you want to continue?”, tekan saja tulisan Confirm
4. Kemudian di halaman selanjutnya, di bagian sudut kanan atas, tekan tombol Create Release. jika berhasil maka aplikasi Anda akan segera online dan bisa di download di Google Play Store.

d. Alat dan Bahan

1. Google Play Developer Console Account

e. Prosedur Kerja

1. Unggah dan Publikasikan aplikasi anda ke Google play Developer Console .
2. Lanjutkan dengan diskusi dengan dosen penguji.

f. Hasil dan Pembahasan

Setiap masukan dari dosen penguji menjadi saran perbaikan project yang dikerjakan.

g. Kesimpulan

Presentasi adalah proses pemaparan hasil penyelesaian final project.

h. Rubrik Penilaian

Penilaian laporan hasil diskusi:

- a. Kelengkapan (25%)
- b. Kesesuaian (25%)
- c. Kedalaman materi (25%)
- d. Kreativitas (25%)

ACARA 46 : BUILD AND RELEASE APLIKASI + PROJECT FINAL

| | |
|----------------------------------|---|
| Pokok Bahasan | : Laporan Studi Kasus |
| Acara Praktikum/Pertemuan | : Minggu 13 / 2 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu menemukan studi kasus yang akan diselesaikan.
2. Mahasiswa mampu mengumpulkan informasi.
3. Mahasiswa mampu menemukan masalah inti dari studi kasus.
4. Mahasiswa mampu menemukan ide untuk solusi dari permasalahan.

b. Indikator

1. Keberhasilan menemukan studi kasus yang akan diselesaikan.
2. Keberhasilan mahasiswa mengumpulkan informasi dan menemukan masalah inti dari studi kasus.
3. Keberhasilan menemukan ide untuk solusi dari permasalahan.

c. Dasar Teori

Design Thinking

Design Thinking adalah proses berulang dimana kita berusaha memahami pengguna, menantang asumsi, dan mendefinisikan kembali masalah dalam upaya mengidentifikasi strategi dan solusi alternatif yang mungkin tidak langsung terlihat dengan tingkat awal pemahaman kita. Pada saat yang sama, Design Thinking menyediakan pendekatan berbasis solusi untuk menyelesaikan masalah. Ini adalah cara berpikir dan bekerja serta kumpulan metode langsung.

Design Thinking berputar di sekitar minat yang mendalam dalam mengembangkan pemahaman dari orang-orang yang menjadi tujuan perancangan produk atau layanan. Hal ini membantu kita mengamati dan mengembangkan empati dengan target pengguna. Design Thinking membantu kita dalam proses bertanya: mempertanyakan masalah, mempertanyakan asumsi, dan mempertanyakan keterkaitannya.

Design Thinking sangat berguna dalam mengatasi masalah-masalah yang tidak jelas atau tidak dikenal, dengan melakukan reframing masalah dengan cara-cara yang berpusat pada manusia, menciptakan banyak ide dalam brainstorming, dan mengadopsi pendekatan langsung dalam pembuatan prototype dan testing. Design Thinking juga melibatkan eksperimen yang sedang berjalan: membuat sketsa, membuat prototype, testing, dan mencoba berbagai konsep dan ide.

Tahapan dalam Proses Design Thinking

Empathise



Tahap pertama dari proses Design Thinking adalah untuk mendapatkan pemahaman empatik tentang masalah yang dicoba untuk diselesaikan. Ini melibatkan para ahli konsultasi untuk mencari tahu lebih banyak tentang bidang yang menjadi perhatian melalui pengamatan, keterlibatan, dan empati dengan orang-orang untuk memahami pengalaman dan motivasi mereka sehingga memperoleh pemahaman pribadi yang lebih jelas tentang masalah yang terlibat. Empati sangat penting untuk proses desain yang berpusat pada manusia seperti Design Thinking, dan empati memungkinkan pemikir desain untuk mengesampingkan asumsi mereka sendiri tentang dunia untuk mendapatkan wawasan tentang pengguna dan kebutuhan mereka.

Define

Selama tahap Define, kita mengumpulkan informasi yang telah kita buat dan kumpulkan selama tahap Empathise. Disinilah kita akan menganalisis pengamatan dan mensistesisnya untuk menentukan masalah inti yang telah diidentifikasi. Kita harus berusaha menidentifikasi masalah sebagai pernyataan masalah dengan cara yang berpusat pada manusia.



Sebagai ilustrasi, alih-alih mengidentifikasi masalah sebagai keinginan atau kebutuhan perusahaan seperti, “Kita perlu meningkatkan pangsa pasar produk makanan diantara remaja perempuan sebesar 5%,” cara yang lebih baik untuk mendefinisikan masalah adalah jadilah, “Gadis remaja perlu makan makanan bergizi agar dapat berkembang, menjadi sehat dan tumbuh.”

Tahap Define akan membantu para desainer dalam sebuah tim untuk mengumpulkan ide-ide hebat untuk membangun fitur, fungsi, dan elemen lain yang akan memungkinkan mereka untuk menyelesaikan masalah atau, paling tidak, memungkinkan pengguna untuk menyelesaikan masalah sendiri dengan tingkat kesulitan minimal.

Ideate



Selama tahap ketiga dari proses Design Thinking, desainer siap untuk mulai menghasilkan ide. Kita telah tumbuh untuk memahami pengguna dan kebutuhan mereka di tahap Empathize, dan kita telah menganalisis dan mensistesis pengamatan Anda di tahap Define, dan berakhir dengan pernyataan masalah yang berpusat pada manusia. Dengan latar belakang yang kuat, kita dan anggota tim dapat mulai “berpikir di luar kotak” untuk mengidentifikasi solusi baru untuk

pernyataan masalah yang dibuat, dan kita dapat mulai mencari cara alternatif untuk melihat masalah.

Ada ratusan teknik Ideation seperti Brainstorm, Brainwrite, Worst Possible Idea, dan SCRAMPER. Sesi Brainstorm dan Worst Possible Idea biasanya digunakan untuk merangsang pemikiran bebas dan untuk memperluas ruang masalah. Penting untuk mendapatkan sebanyak mungkin ide atau solusi masalah. Kita harus memilih beberapa teknik Ideation lainnya pada akhir fase Ideation untuk membantu kita menyelidiki dan menguji ide-ide kita sehingga kita dapat menemukan cara terbaik untuk memecahkan masalah atau menyediakan elemen-elemen yang diperlukan untuk menghindarinya.

<https://sis.binus.ac.id/2020/03/17/design-thinking-pengertian-tahapan-dan-contoh-penerapannya/>

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

1. Mahasiswa berdiskusi dengan kelompok untuk mencari studi kasus dilingkungan sekitar dan menggali informasi serta temukan inti masalah dari setudi kasus.
2. Dari studi kasus yang sudah ditemukan setiap kelompok menganalisa permasalahan inti yang terdapat dalam studi kasus yang ditemukan.
3. Setiap kelompok menyusun pertanyaan untuk pengumpulan data.
4. Setiap kelompok mempresentasikan didepan kelas.
5. Kelompok lain mengoreksi dan mengomentari dari hasil ide yang dikemukakan.

f. Hasil dan Pembahasan

1. Dokumen hasil diskusi kelompok berupa beberapa studi kasus yang diajukan setiap anggota kelompok beserta alasannya sampai dengan hasil pilihan studi kasus yang dipilih, serta beberapa pertanyaan yang dipersiapkan untuk pengumpulan data.
2. Dokumen laporan untuk disusun dalam laporan akhir.

g. Kesimpulan

Mahasiswa menemukan studi kasus yang akan diselesaikan dari hasil diskusi dalam 1 kelompok, dan susunan pertanyaan yang dipersiapkan untuk pengumpulan data

h. Rubrik Penilaian

Penilaian laporan hasil diskusi:

1. Kelengkapan (25%)
2. Kesesuaian (25%)
3. Kedalaman materi (25%)
4. Kreativitas (25%)

ACARA 47 : BUILD AND RELEASE APLIKASI + PROJECT FINAL

| | |
|----------------------------------|---|
| Pokok Bahasan | : Laporan Hasil Observasi |
| Acara Praktikum/Pertemuan | : Minggu 13 / 3 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu menemukan studi kasus yang akan diselesaikan.
2. Mahasiswa mampu mengumpulkan informasi.
3. Mahasiswa mampu menemukan masalah inti dari studi kasus.
4. Mahasiswa mampu menemukan ide untuk solusi dari permasalahan.

b. Indikator

- a. Keberhasilan menemukan studi kasus yang akan diselesaikan.
- b. Keberhasilan mahasiswa mengumpulkan informasi dan menemukan masalah inti dari studi kasus.
- c. Keberhasilan menemukan ide untuk solusi dari permasalahan.

c. Dasar Teori

Proses Bisnis

Definisi Proses Bisnis menurut ahli:

1. **Weske, 2007:** Proses bisnis adalah serangkaian instrumen untuk mengorganisir suatu kegiatan dan untuk meningkatkan pemahaman atas keterkaitan suatu kegiatan.
2. **Sparx System, 2004:** adalah sekumpulan kegiatan atau aktifitas yang dirancang untuk menghasilkan suatu keluaran tertentu bagi pelanggan tertentu.
3. **Hammer dan Champy dalam Weske, 2007:** proses bisnis adalah sekumpulan kegiatan yang mengambil salah satu atau banyak masukan dan menciptakan sebuah keluaran yang berguna bagi pelanggan.

Contoh :

- a. Penerimaan Karyawan oleh HRD
- b. Penjualan Toko oleh Kasir

- c. Pendaftaran Mahasiswa oleh Staff Administrasi
- d. Peminjaman Buku oleh Mahasiswa

Detail Contoh:

- a. Proses bisnis peminjaman buku dapat diuraikan sebagai berikut:
 - a. Memasuki Perpustakaan
 - b. Menunjukkan ID Member
 - c. Memilih Buku
 - d. Membawa buku ke pustakawan
 - e. Memasukkan ID Member dan kode buku
 - f. Melakukan transaksi peminjaman

Teknik Pengumpulan data

- a. **Wawancara** : merupakan suatu teknik pengumpulan data dengan jalan mengadakan komunikasi dengan sumber data. Komunikasi tersebut dilakukan dengan dialog (Tanya jawab) secara lisan, baik langsung maupun tidak langsung
- b. **Observasi** : pengamatan langsung pada lingkungannya dan atau pengamatan langsung suatu kegiatan yang akan dibuat sistemnya.
- c. **Kuesioner dan Literatur** : Teknik yang dilakukan dengan memberikan kuesioner terhadap user yang telah dipilih sedangkan literatur dengan melakukan pencarian terhadap berbagai sumber tertulis, baik berupa buku-buku, arsip, majalah, artikel, dan jurnal, atau dokumen-dokumen yang relevan dengan permasalahan yang dikaji. Sehingga informasi yang didapat dari kuesioner dan studi kepustakaan ini dijadikan rujukan untuk memperkuat system yang akan dibuat.
- d. **Alat dan Bahan**
 - 1. Kertas HVS
 - 2. Spidol Besar
 - 3. Bolpoin
- e. **Prosedur Kerja**

Seluruh anggota kelompok melakukan pengambilan data dan menggali informasi kepada object, dalam bentuk : (1) observasi; (2) wawancara; (3) kuisioner.

- a. Diskusikan dengan masing-masing kelompok hasil pengumpulan data.
- b. Uraikan setiap informasi yang didapatkan dengan bahasa hasil analisa.

- c. Uraikan proses bisnis dari object studi kasus.
- d. Temukan beberapa masalah yang *urgent* dan butuh penyelesaian.
- e. Tentukan masalah yang akan diberikan solusi dari hasil diskusi analisa masalah.

f. Hasil dan Pembahasan

- 1. Dokumen hasil pengumpulan data proses bisnis, permasalahan yang terdapat dalam object serta solusi yang diberikan terhadap masalah yang *urgent*.
- 2. Dokumen laporan untuk disusun dalam laporan akhir.

g. Kesimpulan

Mahasiswa menemukan studi kasus yang akan diselesaikan dari hasil diskusi dalam 1 kelompok.

h. Rubrik Penilaian

Penilaian laporan hasil diskusi:

- a. Kelengkapan (25%)
- b. Kesesuaian (25%)
- c. Kedalaman materi (25%)
- d. Kreativitas (25%)

ACARA 48 : BUILD AND RELEASE APLIKASI + PROJECT FINAL

| | |
|----------------------------------|---|
| Pokok Bahasan | : Laporan Prototype |
| Acara Praktikum/Pertemuan | : Minggu 13 / 4 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

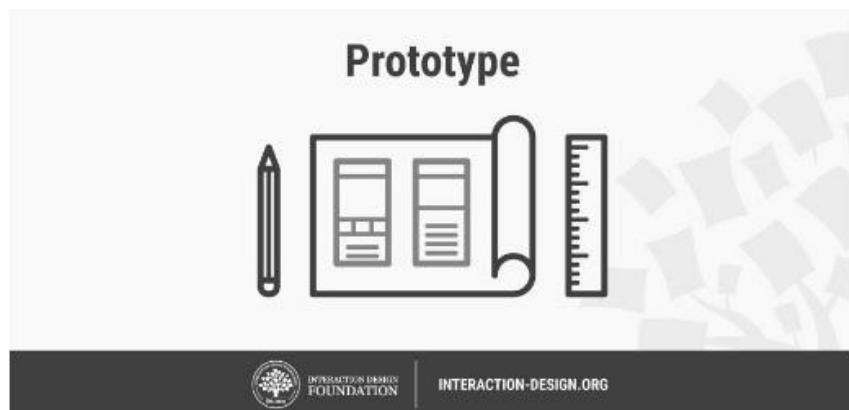
1. Mahasiswa mampu merancang prototype untuk project yang akan dikerjakan.
2. Mahasiswa mampu melakukan validate terhadap prototype.

b. Indikator

Keberhasilan mahasiswa merancang prototype untuk project dan melakukan validate.

c. Dasar Teori

Prototype



Tim desain akan menghasilkan sejumlah versi produk yang murah dan diperkecil atau fitur spesifik yang ditemukan dalam produk, sehingga mereka dapat menyelidiki solusi masalah yang dihasilkan pada tahap sebelumnya. Prototype dapat dibagikan dan diuji dalam tim itu sendiri, di departemen lain, atau pada sekelompok kecil orang diluar tim desain. Ini adalah fase eksperimental, dan tujuannya adalah untuk mengidentifikasi solusi terbaik untuk setiap masalah yang diidentifikasi selama tiga tahap pertama. Solusi diimplementasikan dalam prototype, dan satu per satu, mereka diselidiki dan diterima, diperbaiki dan diperiksa ulang, dan ditolak berdasarkan pengalaman pengguna.

Pada akhir tahap ini, tim desain akan memiliki gagasan yang lebih baik tentang kendala yang melekat pada produk dan masalah yang ada, dan memiliki pandangan yang lebih jelas tentang bagaimana pengguna yang sebenarnya akan berperilaku, berpikir, dan rasakan ketika berinteraksi dengan bagian akhir produk.

Test



Desainer menguji produk lengkap secara ketat menggunakan solusi terbaik yang diidentifikasi selama fase prototyping. Ini adalah tahap akhir dari design thinking, tetapi dalam proses berulang, hasil yang dihasilkan selama fase testing sering digunakan untuk mendefinikan kembali satu atau lebih masalah dan menginformasi pemahaman pengguna, kondisi penggunaan, bagaimana orang berpikir, berperilaku, dan merasakan, dan berempati. Bahkan selama fase ini, perubahan dan penyempurnaan dilakukan untuk menyingkirkan solusi masalah dan memperoleh pemahaman sedalam mungkin terhadap produk dan penggunanya.

<https://sis.binus.ac.id/2020/03/17/design-thinking-pengertian-tahapan-dan-contoh-penerapannya/>

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

1. Berdiskusi dengan masing-masing kelompok untuk merancang prototype project.
2. Membuat sketsa prototype untuk setiap form project yang akan dikerjakan.

3. Implementasikan sketsa ke dalam software untuk menghasilkan prototype yang mudah dipahami.

f. Hasil dan Pembahasan

1. Dokumen hasil pembuatan prototype project akhir menggunakan software yang telah dipilih.
2. Dokumen laporan untuk disusun dalam laporan akhir.

g. Kesimpulan

Mahasiswa dalam 1 kelompok menyelesaikan prototype project akhir yang telah dikerjakan.

h. Rubrik Penilaian

Penilaian laporan hasil diskusi:

2. Kelengkapan (25%)
3. Kesesuaian (25%)
4. Kedalaman materi (25%)
5. Kreativitas (25%)

ACARA 49 : PROJECT FINAL

| | |
|----------------------------------|---|
| Pokok Bahasan | : Laporan Validasi Prototype |
| Acara Praktikum/Pertemuan | : Minggu 14 / 1 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

- Mahasiswa mampu melakukan validasi prototype project yang telah dikerjakan.
- Mahasiswa mampu memperbaiki hasil validasi prototype project.

b. Indikator

Keberhasilan mahasiswa melakukan validasi prototype project dan memperbaikinya.

c. Dasar Teori

-

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

- Diskusi dengan anggota kelompok untuk melakukan validasi prototype.
- Melakukan validasi kepada object studi kasus.
- Diskusi hasil validasi dan memperbaiki prototype jika terdapat ketidak sesuaian dengan keinginan object.
- Mempresentasikan didepan kelas hasil prototype dan hasil perbaikannya.

f. Hasil dan Pembahasan

Dokumentasi hasil validasi prototype dan hasil perbaikan prototype untuk disusun dalam laporan akhir.

g. Kesimpulan

Mahasiswa berhasil melakukan validasi dan perbaikan prototype.

h. Rubrik Penilaian

Penilaian laporan hasil diskusi:

- a. Kelengkapan (25%)
- b. Kesesuaian (25%)
- c. Kedalaman materi (25%)
- d. Kreativitas (25%)

ACARA 50 : PROJECT FINAL

| | |
|----------------------------------|---|
| Pokok Bahasan | : Laporan Usecase Diagram |
| Acara Praktikum/Pertemuan | : Minggu 14 / 2 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu membuat usecase secara kelompok dari studi kasus project akhir.
2. Mahasiswa mampu merancang database secara kelompok dari studi kasus project akhir.

b. Indikator

Keberhasilan mahasiswa membuat usecase, merancang dan menyusun database secara berkelompok dari studi kasus project akhir.

c. Dasar Teori

UML Diagram : Use Case Diagram

Use case diagram merupakan diagram yang menggambarkan hubungan antara aktor dengan sistem. Use case diagram bisa mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Use case diagram juga bisa digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan bisa juga mempresentasikan sebuah interaksi aktor dengan sistem. Komponen tersebut kemudian menjelaskan komunikasi antara aktor, dengan sistem yang ada. Dengan demikian, use case dapat dipresentasikan dengan urutan yang sederhana, dan akan mudah dipahami oleh para konsumen. Manfaat dari use case sendiri adalah untuk memudahkan komunikasi dengan menggunakan domain expert dan juga end user, memberikan kepastian pemahaman yang pas tentang requirement atau juga kebutuhan sebuah sistem.

Use case diagram mempunyai 3 komponen ,yaitu :

Sistem

Menyatakan batasan sistem dalam relasi dengan aktor-aktor yang menggunakannya (di luar sistem) dan fitur-fitur yang harus disediakan (dalam sistem).

Aktor

Aktor adalah segala hal diluar sistem yang akan menggunakan sistem tersebut untuk melakukan sesuatu. Bisa merupakan manusia, sistem, atau device yang memiliki peranan dalam keberhasilan operasi dari sistem.

Use Case

Use Case sendiri adalah gambaran fungsional dari sebuah sistem. Dengan demikian, antara konsumen dan juga pengguna pada sistem tersebut, akan mengerti atau paham mengenai fungsi sistem yang tengah dibangun.

Use case diagram juga mempunyai beberapa relasi, yaitu:

Association

Teknik mengidentifikasi interaksi yang dilakukan oleh actor tertentu dengan use case tertentu. Hal ini digambarkan dengan garis antara aktor terhadap use case tersebut.

Generalization

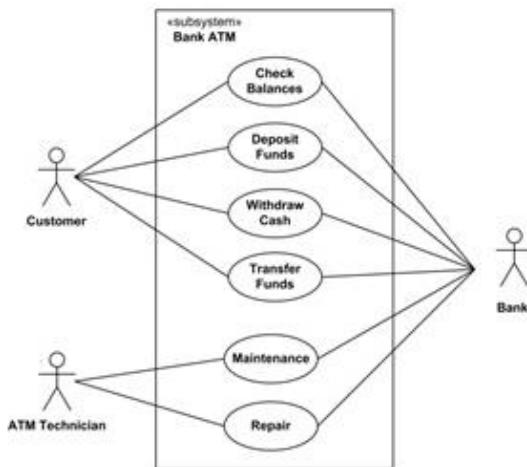
Mendefinisikan relasi antara dua aktor atau dua use case yang mana salah satunya meng-inherit dan menambahkan atau override sifat dari yang lainnya.

Dependency

Dependency ini terbagi menjadi 2 macam, yaitu include dan juga extend.

1. Include : Berfungsi untuk mengidentifikasi hubungan antara 2 use case, dimana use case yang satu akan memanggil use case yang lainnya.
2. Extend : Apabila pemanggilan, memerlukan kondisi tertentu maka akan berlaku dependensi.

Contoh Use Case Diagram :



An example of use case diagram for Bank ATM subsystem - top level use cases.

<https://socs.binus.ac.id/2019/11/26/uml-diagram-use-case-diagram/>

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

Diskusikan dengan kelompok untuk membuat use case project akhir, dan mulai menyusun kedalam laporan akhir.

f. Hasil dan Pembahasan

Dokumentasi use case diagram untuk disusun dalam laporan akhir.

g. Kesimpulan

Mahasiswa membuat use case diagram untuk disusun kedalam project akhir.

h. Rubrik Penilaian

Penilaian laporan hasil diskusi:

- Kelengkapan (25%)
- Kesesuaian (25%)
- Kedalaman materi (25%)
- Kreativitas (25%)

ACARA 51 : PROJECT FINAL

| | |
|----------------------------------|---|
| Pokok Bahasan | : Laporan Database |
| Acara Praktikum/Pertemuan | : Minggu 14 / 3 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu membuat usecase secara kelompok dari studi kasus project akhir.
2. Mahasiswa mampu merancang database secara kelompok dari studi kasus project akhir.

b. Indikator

Keberhasilan mahasiswa membuat usecase, merancang dan menyusun database secara berkelompok dari studi kasus project akhir.

c. Dasar Teori

Definisi Database

Database atau yang dalam bahasa indonesianya dikatakan sebagai basis data/basisdata adalah sekumpulan data yang berupa informasi yang saling berhubungan antara satu dengan lainnya yang disimpan didalam komputer secara sistematis sehingga dapat diakses untuk digunakan kembali.

Pengguna dari database sendiri terbagi menjadi 2 jenis, yaitu database administrator dan database user. Database administrator atau biasa disingkat menjadi DBA berwenang dalam pengawasan database. Selain itu, DBA juga memiliki fungsi untuk mendefinisikan pola struktur database serta memodifikasi pola dan organisasi fisik. Sementara database user adalah pengguna dari database tersebut.

<https://sis.binus.ac.id/2018/12/18/database-secara-umum/>

Conceptual Design Database

Conceptual Design Database adalah tahapan pertama dalam desain database. Tujuan dari tahapan ini adalah untuk merancang database yang independen dari database software dan detail fisik. Tahapan ini akan menghasilkan, conceptual data model yang menjelaskan entitas data, atribut, hubungan antara tabel, dan constraints di suatu database. Design ini bersifat deskriptif dan naratif.

Semua elemen dari data yang dibutuhkan dalam suatu transaksi database harus dijabarkan di modelnya, dan semua elemen data yang ada harus digunakan setidaknya satu kali di suatu transaksi database itu. Conceptual database memiliki 4 tahapan yaitu:

1. Analisis dan persyaratan data
2. Normalisasi dan perancangan hubungan antar entitas
3. Verifikasi data model
4. Distributed database design

<https://sis.binus.ac.id/2020/07/07/database-design-conceptual-design/>

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

Diskusikan dengan kelompok untuk membuat perancangan database project akhir dan meng-implementasip-kan hasil perancangan, dan mulai menyusun kedalam laporan akhir.

f. Hasil dan Pembahasan

Dokumentasi perancangan dan implementasi database untuk disusun dalam laporan akhir.

g. Kesimpulan

Mahasiswa membuat perancangan dan implementasi database untuk disusun kedalam project akhir.

h. Rubrik Penilaian

Penilaian laporan hasil diskusi:

- a. Kelengkapan (25%)
- b. Kesesuaian (25%)
- c. Kedalaman materi (25%)
- d. Kreativitas (25%)

ACARA 52 : PROJECT FINAL

| | |
|----------------------------------|---|
| Pokok Bahasan | : Progres Implementasi Project |
| Acara Praktikum/Pertemuan | : Minggu 14 / 4 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu meng-implementasi-kan aplikasi berbasis android dari kasus tertentu.
2. Mahasiswa mampu menerapkan konsep dasar android yang telah dipelajari.
3. Mahasiswa mampu melaporkan pencapaian progress dari project yang dikerjakan.
4. Mahasiswa mampu mempresentasikan progress masing-masing kelompok.

b. Indikator

Keberhasilan mahasiswa dalam kelompok meng-implementasi-kan aplikasi berbasis android dari kasus tertentu.

c. Dasar Teori

-

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

1. Diskusikan dengan kelompok untuk membuat implementasi aplikasi berbasis android dari kasus tertentu.
2. Push setiap kegiatan dalam membangun aplikasi pada akun GitHub.
3. Invite seluruh anggota kelompok dan dosen pengampu pada GitHub.
4. Melakukan laporan pencapaian dari project yang dikerjakan.
5. Menyampaikan target untuk monitoring selanjutnya.
6. Presentasi progress.

f. Hasil dan Pembahasan

Dokumentasi implementasi project akhir dalam membangun aplikasi pada akun GitHub.

g. Kesimpulan

Mahasiswa setiap kelompok dalam pencapaian progress dari project yang dikerjakan yang di push dalam GitHub.

h. Rubrik Penilaian

Penilaian laporan hasil diskusi:

1. Kelengkapan (25%)
2. Kesesuaian (25%)
3. Kedalaman materi (25%)
4. Kreativitas (25%)

ACARA 53 : PROJECT FINAL

| | |
|----------------------------------|---|
| Pokok Bahasan | : Dokumentasi Project |
| Acara Praktikum/Pertemuan | : Minggu 15 / 1 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu menyusun laporan akhir untuk project akhir.
2. Mahasiswa mampu mendokumentasikan aplikasi dalam bentuk video.

b. Indikator

Keberhasilan mahasiswa setiap kelompok membuat dokumentasi final project berupa laporan dan video.

c. Dasar Teori

-

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

Diskusikan dengan kelompok untuk membuat sebuah video dokumentasi pada aplikasi yang telah dibuat, dan dokumentasi dalam bentuk laporan akhir dan manual book.

f. Hasil dan Pembahasan

- a. Video dokumentasi penggunaan aplikasi dengan durasi maksimal 5 menit.
- b. Laporan akhir dan manual book.

g. Kesimpulan

Mahasiswa setiap kelompok membuat dokumentasi final project berupa video, laporan dan manual book.

h. Rubrik Penilaian

Penilaian laporan hasil diskusi:

1. Kelengkapan (25%)

2. Kesesuaian (25%)
3. Kedalaman materi (25%)
4. Kreativitas (25%)

ACARA 54 : PROJECT FINAL

| | |
|----------------------------------|---|
| Pokok Bahasan | : Presentasi Project Akhir |
| Acara Praktikum/Pertemuan | : Minggu 15 / 2 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

Mahasiswa mampu mempresentasikan hasil projectnya.

b. Indikator

Keberhasilan mahasiswa menjelaskan hasil final project, dan penguasaan materi.

c. Dasar Teori

-

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

- a. Secara bergantian presentasikan hasil penggeraan project, setiap sesi 30 menit.
- b. Lanjutkan dengan diskusi dengan dosen penguji.

f. Hasil dan Pembahasan

Setiap masukan dari dosen penguji menjadi saran perbaikan project yang dikerjakan.

g. Kesimpulan

Presentasi adalah proses pemaparan hasil penyelesaian final project.

h. Rubrik Penilaian

Penilaian laporan hasil diskusi:

- Kelengkapan (25%)
- Kesesuaian (25%)
- Kedalaman materi (25%)
- Kreativitas (25%)

ACARA 55 : PROJECT FINAL

| | |
|----------------------------------|---|
| Pokok Bahasan | : Presentasi Project Akhir |
| Acara Praktikum/Pertemuan | : Minggu 15 / 3 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

Mahasiswa mampu mempresentasikan hasil projectnya.

b. Indikator

Keberhasilan mahasiswa menjelaskan hasil final project, dan penguasaan materi.

c. Dasar Teori

-

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

- a. Secara bergantian presentasikan hasil penggeraan project, setiap sesi 30 menit.
- b. Lanjutkan dengan diskusi dengan dosen penguji.

f. Hasil dan Pembahasan

Setiap masukan dari dosen penguji menjadi saran perbaikan project yang dikerjakan.

g. Kesimpulan

Presentasi adalah proses pemaparan hasil penyelesaian final project.

h. Rubrik Penilaian

Penilaian laporan hasil diskusi:

- a. Kelengkapan (25%)
- b. Kesesuaian (25%)
- c. Kedalaman materi (25%)
- d. Kreativitas (25%)

ACARA 54 : PROJECT FINAL

| | |
|----------------------------------|---|
| Pokok Bahasan | : Presentasi Project Akhir |
| Acara Praktikum/Pertemuan | : Minggu 15 / 4 |
| Tempat | : Daring / Luring Politeknik Negeri Jember |
| Alokasi Waktu | : 100 menit |

a. Capaian Pembelajaran Mata Kuliah (CPMK)

Mahasiswa mampu mempresentasikan hasil projectnya.

b. Indikator

Keberhasilan mahasiswa menjelaskan hasil final project, dan penguasaan materi.

c. Dasar Teori

-

d. Alat dan Bahan

1. Laptop
2. LCD proyektor

e. Prosedur Kerja

- a. Secara bergantian presentasikan hasil penggeraan project, setiap sesi 30 menit.
- b. Lanjutkan dengan diskusi dengan dosen penguji.

f. Hasil dan Pembahasan

Setiap masukan dari dosen penguji menjadi saran perbaikan project yang dikerjakan.

g. Kesimpulan

Presentasi adalah proses pemaparan hasil penyelesaian final project.

h. Rubrik Penilaian

Penilaian laporan hasil diskusi:

- a. Kelengkapan (25%)
- b. Kesesuaian (25%)
- c. Kedalaman materi (25%)
- d. Kreativitas (25%)