

berikut perbedaan antara Development Native dan Hybrid :

#### 1. Native Development:

- Bahasa Pemrograman: Aplikasi dikodekan menggunakan bahasa pemrograman yang spesifik untuk platform tertentu, seperti Java atau Kotlin untuk Android, dan Swift atau Objective-C untuk iOS.

- Framework dan API Platform: Menggunakan framework dan API platform yang disediakan oleh vendor, seperti Android SDK atau iOS SDK. Ini memberikan akses langsung ke fitur-fitur platform.

Performa: Biasanya memberikan performa terbaik karena aplikasi dikompilasi secara langsung untuk platform target.

- User Experience: Memungkinkan untuk memberikan pengalaman pengguna yang lebih baik karena aplikasi dapat menyesuaikan diri sepenuhnya dengan standar UI/UX platform.

- Fitur Platform: Dapat mengakses semua fitur dan fungsionalitas platform secara penuh.

Kompleksitas Pengembangan: Memerlukan pengembangan terpisah untuk setiap platform, sehingga membutuhkan waktu dan usaha yang lebih besar.

#### Hybrid Development:

- Bahasa Pemrograman: Menggunakan teknologi web standar seperti HTML, CSS, dan JavaScript.

Framework Cross-Platform: Menggunakan kerangka kerja seperti React Native, Flutter, atau Xamarin yang memungkinkan pengembangan satu kode yang berjalan di beberapa platform.

- Performa: Performa mungkin tidak sebaik aplikasi native karena aplikasi hybrid harus dijalankan di dalam wrapper native dan menggunakan mekanisme untuk berkomunikasi dengan komponen platform.

- User Experience: Meskipun kerangka kerja hybrid berusaha untuk menyamai tampilan dan perilaku aplikasi native, beberapa perbedaan mungkin terlihat dalam hal responsif, animasi, dan integrasi platform.

- Fitur Platform: Perlu memanfaatkan plugin atau ekstensi khusus untuk mengakses fitur-fitur platform tertentu.

- Kompleksitas Pengembangan: Mengurangi kompleksitas dengan mengizinkan pengembangan satu kode untuk beberapa platform, tetapi masih memerlukan penanganan

kasus khusus untuk setiap platform.

Keputusan antara pengembangan natif dan hibrida seringkali tergantung pada kebutuhan proyek, anggaran, waktu, dan sumber daya yang tersedia. Aplikasi yang membutuhkan performa tinggi atau memiliki integrasi platform yang kompleks cenderung lebih cocok untuk pengembangan natif, sementara aplikasi dengan jadwal yang ketat atau anggaran yang terbatas mungkin lebih cocok untuk pendekatan hibrida.