

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Tinjauan Pustaka**

Pada penelitian ini digunakan tinjauan studi yang nantinya akan mendukung penelitian, berikut ini merupakan tinjauan studi yang diambil oleh penulis sebagai berikut :

##### **2.1.1. Tinjauan Literatur 1**

Pada penelitian literatur 1 dengan judul Aplikasi Pemilihan Jasa Tukang Menggunakan Metode *Weighted Product* Berbasis Web pada tahun 2019 oleh Y Setiawan. Aplikasi tersebut meliputi sebuah sistem yang memberikan informasi mengenai jasa tukang. Sehingga aplikasi ini dapat membantu masyarakat dalam mencari atau menggunakan jasa tukang tanpa takut harga mahal dan kinerja yang tidak bagus. Aplikasi ini dapat mempermudah masyarakat yogyakarta dalam memilih jasa tukang dan melihat rekomendasi jasa tukang yang sudah disarankan berdasarkan ongkos dan rating. Lalu berdasarkan hasil pengamatan dan penelitian yang telah dilakukan, dapat disimpulkan bahwa dengan menggunakan metode *weighted product* pada aplikasi ini dapat mempermudah masyarakat yogyakarta dalam memilih jasa tukang. Kemudian metode *weighted product* yang digunakan sistem mampu melakukan pengurutan jasa tukang sebagai hasil rekomendasi jasa tukang yang disarankan berdasarkan ongkos dan rating.

##### **2.1.2. Tinjauan Literatur 2**

Pada penelitian literatur 2 dengan judul Aplikasi Pemesanan Jasa Tukang Menggunakan Teknologi *Webhook* dan *Google Event Calendar* pada tahun 2019 oleh Muhammad Nafis. Jasa tukang adalah sebuah layanan publik yang merupakan

orang yang mempunyai kepandaian dalam suatu pekerjaan, juga merupakan sebuah kegiatan dan manfaat yang di tawarkan oleh pihak penyedia jasa ke pihak pengguna jasa. Belum ada solusi mudah untuk proses pemesanan jasa tukang yang sesuai keahlian dan permintaan yang dibutuhkan, oleh karena itu dibutuhkan suatu wadah untuk mengatasi permasalahan tersebut yaitu dibuatkan aplikasi yang bernama Jakang. Aplikasi Jakang ini memiliki fungsi untuk mempermudah penggunaan jasa dalam memesan jasa tukang sesuai keahlian dan sesuai permintaan yang dibutuhkan oleh rekomendasi sistem. Adapun teknologi yang digunakan adalah *webhook* sebagai notifikasi otomatis sebelum pengerjaan, dan juga *google event calendar* digunakan untuk proses penjadwalan.

### **2.1.3. Tinjauan Literatur 3**

Pada penelitian literatur 3 dengan judul Rancang Bangun Sistem Rekomendasi Pariwisata Mobile dengan Menggunakan Metode *Collaborative Filtering* dan *Location Based Filtering* pada tahun 2019 oleh Assaf Arief. Penelitian ini bertujuan untuk membuat otomatis sistem rekomendasi pariwisata *mobile* menggunakan konsep personalisasi dengan metode *collaborative filtering* dan *location based filtering* sebagai pengganti dari sistem manual ( seorang pemandu). Langkah – langkah menyelesaikan penelitian ini adalah membuat sistem rekomendasi pariwisata *mobile* dengan metode tersebut dan membuat *prototype* aplikasi berbasis *mobile web*, melakukan perancangan, pembuatan, pengujian dan perbaikan sistem rekomendasi. Hasil penelitian ini berupa sebuah aplikasi sistem rekomendasi pariwisata berbasis *mobile web* menggunakan *platform query mobile*, HTML 5, javascript, Ajax, PHP dan MySQL. Dari proses pengujian fungsionalitas,

pengujian komparabilitas dan pengujian sistem rekomendasi dapat terlihat bahwa rancangan sistem rekomendasi pariwisata sudah dapat memberikan rekomendasi wisata sesuai dengan metode *Collaborative Filtering* dan *Location-based filtering* yang dirancang.

#### **2.1.4. Tinjauan Literatur 4**

Pada penelitian literatur 4 dengan judul Sistem Rekomendasi dengan Metode *Item-Based Collaborative Filtering* Pada Aplikasi *Mobile* Berbasis Android untuk Wisata Kuliner Kota Surakarta pada tahun 2019 oleh Vina Zahrotun. Penelitian ini bertujuan membangun sebuah sistem rekomendasi dengan *item-based collaborative filtering* yang diterapkan dalam teknologi *mobile*. Proses utama rekomendasi dalam sistem ini menggunakan item-based CF dengan 5 item tetangga terdekat (*neighborhood*) karena hasil analisis pengujian *MAE* dengan pengosongan acak sebesar 5%, 15%, 10%, dan 20% dengan variasi jumlah *neighborhood* dan menunjukkan bahwa metode *item-based collaborative filtering* dengan 5 tetangga dapat menghasilkan prediksi dengan kualitas akurasi yang baik.

#### **2.1.5. Tinjauan Literatur 5**

Pada penelitian literatur 5 dengan judul Aplikasi Rekomendasi Tempat Makan Menggunakan Algoritma *Slope One* pada Platform Android pada tahun 2020 oleh Dharma Pratama. Pada aplikasi ini, rekomendasi yang diberikan didapatkan dari perhitungan menggunakan algoritma *slope one* dan daftar tempat makan yang didapatkan berasal dari *Google Places API*. Algoritma *slope one* membuat rekomendasi dengan menjumlahkan rating dari suatu tempat dengan rata-

rata selisih suatu tempat dengan tempat lainnya. Aplikasi telah diuji coba pada pengguna menggunakan kuisioner yang dibuat berdasarkan kuisioner J.R. Lewis dengan katagori pertanyaan tingkat kegunaan aplikasi, kualitas informasi yang diberikan oleh aplikasi dan kualitas tampilan antar muka aplikasi. Hasil yang didapat dari uji coba melalui kuisioner adalah aplikasi berguna bagi user untuk menentukan pilihan tempat makan, kualitas informasi yang diberikan aplikasi baik dan tampilan antar muka dari aplikasi baik, yang berarti pengguna merasa puas dalam menggunakan aplikasi.

#### **2.1.6. Kesimpulan Tinjauan Literatur**

Kesimpulan yang dapat penulis ambil dari tinjauan literatur bahwa sistem rekomendasi dapat dilakukan dengan beberapa metode. Salah satu metode rekomendasi yaitu *item-based collaborative filtering* dinyatakan mampu memberikan hasil prediksi dengan kualitas akurasi yang baik dan telah di uji pada tinjauan literatur 4. Algoritma *Slope one* dinyatakan mampu membuat rekomendasi dengan menjumlah rating dari suatu tempat dengan rata-rata selisih suatu tempat dengan tempat yang lainnya dan telah di uji pada tinjauan literatur 5. Maka dari itu penulis menggunakan metode dan algoritma tersebut yaitu Metode *Collaborative Filtering* dan algoritma *Slope One* dalam penelitian ini dan untuk membangun sebuah aplikasi jasa tukang seperti pada tinjauan literatur 1 dan 2 dengan memasukan sistem rekomendasi. Perbedaan yang terdapat pada penelitian penulis adalah aplikasi jasa tukang yang akan dibuat oleh penulis berbasis android dan memberikan sistem rekomendasi untuk memilih jasa tukang di dalamnya.

## 2.2. Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Pada saat perilisan perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan *open source* pada perangkat *mobile* (M Nafis, 2019). Di lain pihak, google merilis kode-kode android dibawah lisensi Apache, sebuah lisensi perangkat lunak *open platform* perangkat seluler .

Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat full support dari Google atau *Google Mail Service (GMS)* dan kedua tanpa dukungan Google atau *Open Handset Distribution (OHD)*. Pada masa saat ini kebanyakan vendor-vendor smartphone sudah memproduksi smartphone berbasis android, diantaranya di indonesia adalah Xiaomi, Samsung, Asus, Vivo, Oppo dan masih banyak lagi vendor yang memproduksi Andorid.

Pesatnya pertumbuhan Android selain faktor diatas adalah karena Android itu sendiri adalah *platform* yang sangat lengkap baik itu sistem operasinya, Aplikasinya dan *Tool Pengembangan, Market* serta dukungan sangat tinggi dari komunitas *Open Source* di dunia, sehingga android terus berkembang sangat pesat baik dari teknologi maupun dari segi jumlah *device* yang ada di dunia. Berikut adalah logo android yang ditunjukkan pada gambar 2.1 .



**Gambar 2.1** Logo Android

### 2.3. Android Studio

Android Studio merupakan sebuah *Integrated Development Environment (IDE)* khusus untuk membangun aplikasi yang berjalan pada platform android. Android studio ini berbasis pada *IntelliJ IDEA*, sebuah *IDE* untuk bahasa pemrograman *Java*. Bahasa pemrograman utama yang digunakan adalah *java*, sedangkan untuk membuat tampilan atau *layout*, digunakan bahasa *XML*. Android studio juga terintegrasi dengan *Android Software Development Kit (SDK)* untuk *deploy* ke perangkat android. Android studio juga merupakan pengembangan dari *eclipse*, dikembangkan menjadi lebih kompleks dan professional yang telah tersedia didalamnya *Android Studio IDE*. (Yulianti dan Dita, 2019).



**Gambar 2.2** Logo Android Studio

Setiap proyek di Android Studio berisi satu atau beberapa modul dengan file kode sumber dan file sumber daya. Jenis-jenis modul mencakup :

- a. Modul aplikasi android
- b. Modul Pustaka
- c. Modul *Google App Engine*

Secara default, Android studio akan menampilkan file proyek dalam tampilan proyek android.

## 2.4. Algoritma

Algoritma adalah kumpulan instruksi atau perintah yang dibuat secara jelas dan sistematis berdasarkan urutan yang logis (logika) untuk penyelesaian suatu masalah. French, C.S (2020) menyatakan sejumlah konsep yang mempunyai relevansi dengan masalah rancangan program yaitu kemampuan komputer, kesulitan dan ketepatan. Knuth (2020) menyatakan algoritma fundamental untuk keperluan matematika dan program komputer.

Algoritma tidak selalu memberikan hasil terbaik yang mungkin diperoleh, maka diharapkan adanya suatu evaluasi mutu hasil dari algoritma tersebut (Liu, C.L, 2019). Sekali sebuah algoritma diberikan kepada sebuah permasalahan dan dijamin akan memberikan hasil yang diharapkan, maka langkah penting selanjutnya adalah menentukan besar biaya yang diperlukan algoritma tersebut untuk memperoleh hasil.

Dalam dunia komputer, algoritma sangat berperan penting dalam pembangunan suatu *software*. Dalam dunia sehari-hari, mungkin tanpa kita sadari algoritma telah masuk dalam kehidupan kita seperti cara memasak, membuat teh, cara membuat kopi dan lain-lain.

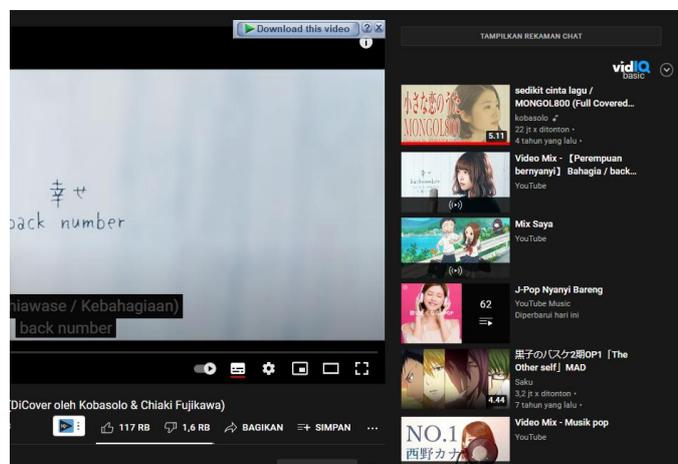
## 2.5. Sistem Rekomendasi

Sistem rekomendasi merupakan suatu sistem yang dirancang untuk memprediksi sekumpulan *item* yang sesuai dengan preferensi *user* yang mana nantinya item tersebut akan direkomendasikan pada *user* lain (Sanjung, 2019). Perkiraan informasi ini diperoleh melalui profil *user*, preferensi *item* dan aktivitas yang terjadi pada sistem. Profil *user* dapat berisi tentang informasi *user*,

ketertarikan *user* pada suatu *item* dan juga riwayat interaksi antara pengguna dengan *item*. Misalnya ketika pengguna melakukan peminjaman buku maka data riwayat peminjaman ini akan dicatat dan menjadi profil pengguna.

Pada sistem rekomendasi terdapat sekumpulan *item* yang ada akan disaring berdasarkan preferensi *item*, *user*, *rating*, catatan *history* transaksi dan lain lain sehingga menghasilkan beberapa *item* yang direkomendasikan kepada *user*. Pada saat ini sudah banyak yang menerapkan sistem rekomendasi pada *website* untuk mempermudah *user* mencari informasi, seperti youtube.com dan lain lain.

Gambar 2.3 merupakan *screenshot* dari halaman situs youtube.com. Pada halaman tersebut dapat dilihat pada sisi sebelah kanan terdapat beberapa video yang direkomendasikan kepada *user* berdasarkan *history* pencarian yang pernah dilakukan *user* tersebut.



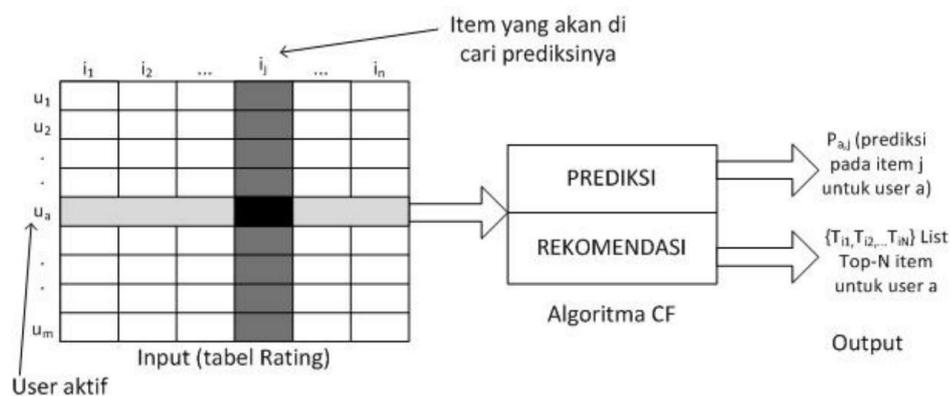
**Gambar 2.3** Rekomendasi Youtube

## 2.6. Collaborative Filtering

*Collaborative Filtering* merupakan salah satu cara yang diterapkan sistem rekomendasi untuk memberikan prediksi otomatis terhadap keinginan pelanggan dengan cara mengkoleksi informasi dari banyak pelanggan ( Ampaziz, 2020). Pada

*collaborative filtering*, rekomendasi yang diberikan berdasarkan keterkaitan (preferensi) antara *item* satu dengan *item* lainnya dan keterkaitan yang didapat diantara *user* yang telah *me-rating* atau berinteraksi dalam sistem.

*Collaborative filtering* memberikan rekomendasi berdasarkan kumpulan dari pendapat, minat dan keterkaitan beberapa *user* yang biasanya diberikan dalam bentuk *rating* yang diberikan *user* kepada suatu *item*. Pada metode ini, terdapat kumpulan  $m$  *user*  $U = \{u_1, u_2, u_3, \dots\}$  dan kumpulan  $n$  *item*  $I = \{i_1, i_2, i_3, \dots\}$  dimana setiap *user* ( $u_i$ ) mempunyai daftar *item*  $I_{u_i}$  yang merupakan ekspresi dari pendapatnya.



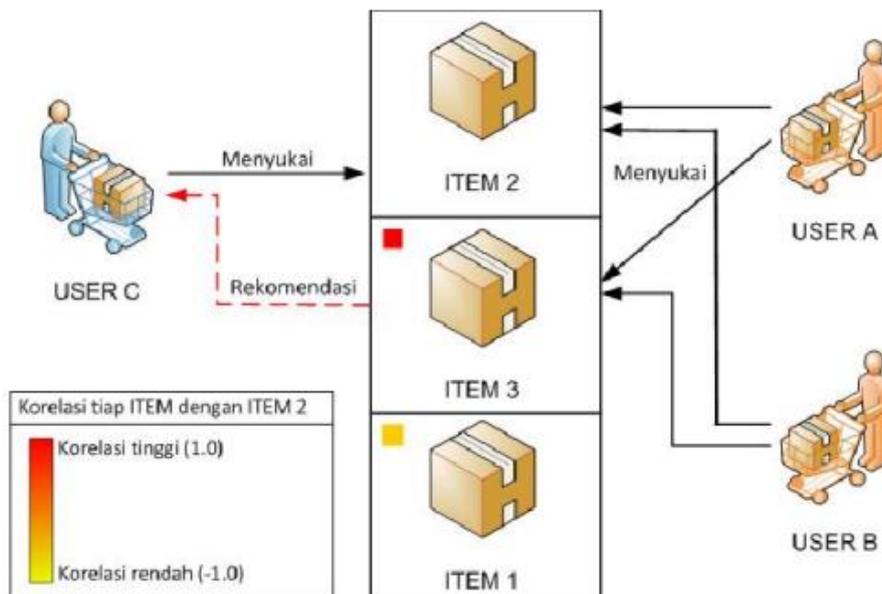
**Gambar 2.4** Proses *Collaborative Filtering* ( Hakim, 2020)

Gambar 2.4 diatas merupakan diagram skema dari metode *collaborative filtering*. Pada gambar tersebut dipresentasikan sejumlah  $m$  *user* x  $n$  *item* sebagai matriks rating dimana berisi nilai *rating* dari *user* untuk setiap *item*.

Pendekatan *collaborative filtering* pada dasarnya dibagi menjadi dua katagori yaitu *userbased collaborative filtering* disebut juga *memorybased*, dan *itembased collaborative filtering* yang disebut juga *model-based* ( Uyun, 2019). Pada pendekatan *userbased collaborative filtering* sistem memberikan rekomendasi kepada *user* *item-item* yang disukai atau di rating oleh *user-user* lain

yang memiliki banyak kemiripan dengannya. Misalnya, *user a* menyukai atau merating *item 1, 2, dan 3*, kemudian *user b* menyukai *item 1, 2, dan 4* maka sistem akan merekomendasikan *item 3* kepada *user b* dan *item 4* kepada *user a*. Kelebihan dari pendekatan *user based collaborative filtering* adalah dapat menghasilkan rekomendasi yang berkualitas baik. Sedangkan kekurangannya adalah kompleksitas perhitungan akan semakin bertambah seiring dengan bertambahnya pengguna sistem, semakin banyak pengguna yang menggunakan sistem, maka proses perkomendasi akan semakin lama.

Pendekatan *item based collaborative filtering* memberikan rekomendasi berdasarkan kemiripan antar *item*. Metode ini merupakan metode rekomendasi yang didasari atas adanya kesamaan antara pemberian *rating* terhadap suatu item dengan *item* yang pernah di-*rating user* lain (Purwanto, 2020). *Item* yang telah di *rating* oleh *user* akan menjadi patokan untuk mencari sejumlah *item* lainnya yang berkorelasi dengan *item* yang telah di *rating user*. Motivasi kunci dibalik metode ini adalah *user* akan cenderung menyukai *item* yang sejenis atau mempunyai korelasi dengan *item* yang telah disukainya. (Sarwar, 2019). Pada gambar 2.5 menggambarkan bagaimana *item 3* direkomendasikan kepada *user c* yang telah menyukai *item 2*, dimana *item 2* yang disukai *user c* memiliki korelasi yang kuat dengan *item 3*.



**Gambar 2.5** Skema *item-based collaborative filtering* (Hakim, 2020)

## 2.7. Algoritma *Slope One*

Algoritma *slope one* adalah salah satu algoritma untuk membuat sistem rekomendasi. *Slope one* memberikan prediksi berdasarkan nilai hasil pencarian dari item-item yang dibandingkan. Keunggulan algoritma ini dibandingkan dengan algoritma lainnya adalah algoritma *slope one* mudah untuk diimplementasi, efisien saat melakukan query, tidak memerlukan banyak requirement dikarenakan rekomendasi berdasarkan rating dari setiap *item*, dan cukup akurat.

Algoritma *slope one* melakukan perhitungan berdasarkan hubungan linear dari nilai preferensi atau *weight* dari setiap *item* yang dibandingkan. Estimasi umum dari dasar perhitungan algoritma ini adalah fungsi linear  $y = mx + b$ , dengan asumsi *gradient*  $m=1$ , sehingga fungsi menjadi  $b = y - x$ . cara kerja algoritma *slope one* adalah dengan mencari selisih dari suatu *item* dengan *item-item* lain yang

dibandingkan. Perhitungan algoritma *slope one* dapat diformulasikan dengan persamaan dengan

$$dev_{j,i} = \sum_{u \in S_{j,i}(x)} \frac{u_j - u_i}{card(S_{j,i}(x))}$$

Dimana :

$dev_{j,i}$  = rata-rata selisih rating item j dan i

$u_j$  = rating item j

$u_i$  = rating item i

$card(S_{j,i}(x))$  = banyaknya elemen yang dibandingkan

Apabila selisih sudah didapatkan, maka dapat dilakukan perhitungan rekomendasi untuk item j yang dapat dirumuskan dengan persamaan :

$$p^{SI}(u)_j = dev_{j,i} + u_j$$

Dimana :

$p^{SI}(u)_j$  = nilai rekomendasi untuk item j

Berdasarkan persamaan di atas, algoritma *slope one* memberikan rekomendasi dengan melakukan perhitungan selisih setiap item. Selisih yang didapatkan akan dirata-ratakan per *item* yang kemudian akan dijumlahkan dengan *value* dari masing-masing *item*. *Value* yang sudah dijumlahkan dengan rata-rata selisihnya akan digunakan sebagai point untuk memberikan rekomendasi. Gambar 2.6 memperlihatkan *pseudocode* algoritma *slope one*.

```

BEGIN
  For every item i
    For every other item j
      For every user u expressing preference for
        both i and j
        Add the difference in u's preference for
          i and j to an average
      //Prediction Part
      For every item i the user u expresses no
        preference for
        For every item j that user u expresses a
          preference for
          Find the average preference difference
            between j and i
          Add this diff to u's preference value for
            j
          Add this to a running average
      Return the top items, ranked by these averages
END

```

**Gambar 2.6** *Pseudocode slope one*

## 2.8. Aplikasi

Aplikasi merupakan perangkat lunak yang dijalankan oleh para pengguna atau biasa disebut dengan user untuk mendapatkan suatu tujuan tertentu. Aplikasi perangkat lunak adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk bertujuan melakukan suatu tugas yang diinginkan pengguna. Aplikasi perangkat lunak adalah program yang membuat komputer dapat digunakan untuk pekerjaan sehari-hari agar lebih efektif dan efisien.

Biasanya dibandingkan dengan perangkat lunak sistem yang menintegrasikan berbagai kemampuan komputer tetapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan. Program ini lah yang mengendalikan semua aktifitas yang ada pada pemroses. Program berisi konstruksi logika yang ada dibuat oleh manusia, dan sudah diterjemahkan ke dalam bahasa mesin sesuai dengan format yang ada pada *instuction set*. Program aplikasi merupakan program siap pakai yang dirancang untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi lain.

Contoh-contoh aplikasi ialah program pemroses kata dan *web browser*. Aplikasi akan menggunakan sistem operasi komputer dan aplikasi yang lainnya yang mendukung. Istilah ini mulai perlahan masuk ke dalam istilah Teknologi informasi.

## 2.9. Firebase

Firebase memiliki produk utama, yaitu menyediakan *database realtime* dan *backend* sebagai layanan. Layanan ini menyediakan pengembangan aplikasi *API* yang memungkinkan aplikasi data yang akan disinkronasikan di klien dan disimpan di cloud firebase ini. Firebase menyediakan library untuk berbagai *client platform* yang memungkinkan integrasi dengan Android, IOS, Javascript Node Js dan dapat juga disebut sebagai layanan *DbaaS (Database as a Service)* dengan konsep *realtime*.

Firebase digunakan untuk mempermudah dalam penambahan fitur – fitur yang akan dibangun oleh *developer*. Semua data firebase *realtime database* disimpan sebagai objek JSON. Tidak seperti basis data SQL, tidak ada tabel atau rekaman. Ketika ditambahkan ke JSON tree, data akan menjadi simpul dalam struktur JSON yang ada. Meskipun basis data menggunakan JSON tree, data yang tersimpan dalam basis data bisa diwakili sebagai tipe bawaan tertentu yang sesuai dengan tipe JSON yang tersedia untuk membantu menulis banyak kode.

## 2.10. Tukang

Terdapat 5 arti kata “Tukang” di Kamus Besar Bahasa Indonesia (KBBI) yang masuk kedalam kelas verba (kata kerja)

1. Orang yang mempunyai kepandaian dalam suatu pekerjaan tangan ( dengan alat atau bahan yang tertentu).

Contoh : batu, besi, kayu

2. Orang yang pekerjaannya membuat (menjual, memperbaiki, dan sebagainya)

Contoh : daging (sayur,susu), arloji, pedati (gerobak, becak)

3. Orang yang pekerjaannya melakukan sesuatu secara tetap

Contoh : pangkas (cukur), las, jahit, masak, cetak

4. Orang yang biasa suka melakukan sesuatu (yang kurang baik)

Contoh : mabuk, serabot, copet, tadah, catut

5. Ahli dalam ragam cakapan

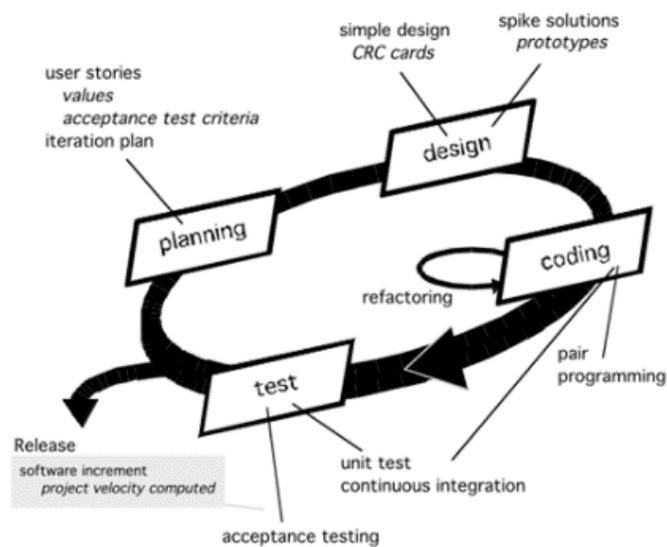
Contoh : menciptakan sajak, pidato

### **2.11. Bangunan**

Menurut Kamus Besar Bahasa Indonesia (KBBI) arti kata bangunan adalah bangunan yang didirikan, yang dibangun (seperti rumah, gedung, jembatan). Arti lainnya jika ditambah “—liar” adalah bangunan yang didirikan secara tidak sah tanpa memperoleh izin membangun atau yang didirikan di atas tanah bukan milik sendiri. Arti lain dengan tambahan “-permanen” adalah bangunan yang dibuat dengan bahan bangunan yang kuat dan tahan lama seperti dari baja, beton, batu bata

### 2.12. *Extereme Programming*

*Extreme Programming (XP)* merupakan sebuah proses rekayasa perangkat lunak yang cenderung menggunakan pendekatan berorientasi objek dan sasaran dari metode ini adalah tim yang dibentuk dalam skala kecil sampai medium serta metode ini juga sesuai jika tim dihadapkan dengan *requirement* yang tidak jelas maupun terjadi perubahan *requirement* yang cepat. Ilustrasi XP dapat dilihat pada gambar 2.3



**Gambar 2.7** Extreme Programming

Adapun tahapan dalam XP ini adalah sebagai berikut :

1. *Planning* (Perencanaan)

Tahapan ini dimulai dengan mendengarkan kumpulan kebutuhan aktifitas suatu sistem yang memungkinkan pengguna memahami proses bisnis untuk sistem dan mendapatkan gambaran yang jelas mengenai fitur utama, fungsionalitas dan keluaran yang di inginkan

## 2. *Design* (Perancangan)

Pada tahapan perancangan dilakukan pembuatan pemodelan sistem berdasarkan hasil analisa kebutuhan yang didapatkan. Selain itu dibuatkan juga pemodelan basis data untuk menggambarkan hubungan antar data. Pemodelan sistem yang digunakan yaitu *Simple Design CRC Card* dan *Spike solutions prototypes*

## 3. *Coding* (Pengkodean)

Tahapan ini merupakan implementasi dari perancangan model sistem yang telah dibuat kedalam kode program yang menghasilkan prototipe dari perangkat lunak.

## 4. *Testing* (Pengujian)

Tahapan ini merupakan tahapan pengujian terhadap aplikasi yang sudah dibangun, pada tahapan ini ditentukan oleh pengguna sistem dan berfokus pada fitur dan fungsionalitas dari keseluruhan sistem.

## 5. Software increment (Peningkatan Perangkat Lunak)

Tahapan ini merupakan tahap pengembang sistem yang sudah dibuat secara bertahap yang dilakukan setelah sistem diterapkan dalam organisasi dengan menambah layanan atau konten yang mengakibatkan bertambahnya kemampuan fungsionalitas dari sistem

### **2.13. Mobile Application**

*Mobile application* adalah sebuah aplikasi yang memungkinkan manusia melakukan mobilitas dengan menggunakan perlengkapan seperti PDA, telepon seluler (*handphone*). Dengan menggunakan aplikasi *mobile*, manusia dapat dengan mudah melakukan berbagai macam aktifitas mulai dari hiburan, berjualan, belajar,

mengerjakan pekerjaan kantor, browsing, chatting, e-mail dan sebagainya. (Yonarisa, 2019)

#### **2.14. Android SDK**

Android SDK (*software development kit*) merupakan *tools* bagi para programmer yang ingin mengembangkan aplikasi berbasis google android. Android SDK mencakup seperangkat alat pengembangan yang komprehensif. Android SDK terdiri dari *debugger, libraries, handset emulator*, dokumentasi, contoh kode, dan tutorial (Onserda, 2020)

#### **2.15. IDE (Integrated Development Environment)**

Adalah program komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat lunak. Tujuan dari IDE adalah untuk menyediakan semua utilitas yang diperlukan dalam membangun perangkat lunak. Sebuah IDE atau secara bebas dapat diterjemahkan sebagai lingkungan pengembangan terpadu. Berikut beberapa fasilitas yang diberikan oleh IDE :

1. *Editor*, yaitu fasilitas untuk menulis kode sumber dari perangkat lunak.
2. *Compiler*, yaitu fasilitas untuk mengecek sintaks dari kode sumber kemudian mengubahh dalam bentuk binari yang sesuai dengan bahasa mesin.
3. *Linker*, yaitu fasilitas untuk menyatukan data binari dari beberapa kode sumber yang dihasilkan compiler sehingga data-data binari tersebut menjadi satu kesatuan dan menjadi suatu program siap dieksekusi.
4. *Debugger*, yaitu fasilitas yang mengetes jalannya program untuk mencari *bug* / kesalahan yang terdapat pada program.