

BAB III

PELAKSANAAN KERJA MAGANG

3.1. Kedudukan dan Koordinasi

Tugas yang dilakukan penulis selama kerja magang di PT Suitmedia Kreasi Indonesia adalah melakukan pengembangan sistem *omnichannel* PT Ritel dan Multi Manufaktur bagian *frontend* program loyalitas pelanggan. Penulis bekerja pada divisi frontend developer di bawah bimbingan Bapak Andhika Ridhovani selaku Technical Lead Frontend Developer atas keputusan dari Bapak Chandra Rolexta selaku Technical Manager divisi Frontend Developer.

Penulis berkoordinasi dengan divisi backend untuk proses integrasi data, Project Manager PT Suitmedia Kreasi Indonesia, Project Manager PT Ritel dan Multi Manufaktur, serta tim analis untuk melakukan *testing* dan *bug fixing*.

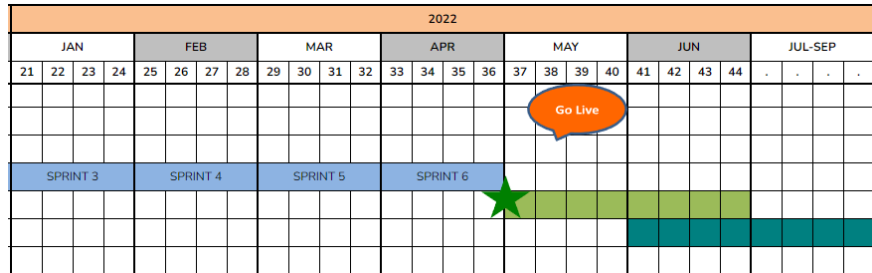
3.2. Tugas dan Uraian Kerja Magang

3.2.1. Tugas Kerja Magang

Dua minggu awal kerja magang penulis melakukan studi dokumentasi *boilerplate framework* divisi frontend PT Suitmedia Kreasi Indonesia dan melakukan beberapa proyek percobaan yang ditentukan oleh Technical Lead Frontend Developer. Adapun basis dari *boilerplate* yang digunakan adalah *Nuxt JS* yang telah ditambahkan modul *styling*, *middleware*, *plugins*, dan *dependencies* lain yang diperlukan untuk membangun website yang sesuai dengan permintaan *client*.

Selanjutnya penulis bergabung dalam tim pengembangan sistem *omnichannel* untuk PT Ritel dan Multi Manufaktur pada bagian *website* program loyalitas pelanggan. Dalam proses

pengembangannya, digunakan metode scrum untuk manajemen tugas pengembangan. Dimana dalam satu siklusnya (*sprint*) mempunyai waktu sekitar dua minggu.



Gambar 3. 1. Linimasa Proyek Sistem Omnichannel PT Ritel dan Multi Manufaktur

Berikut adalah detail linimasa per minggu kegiatan yang dilakukan penulis selama kerja magang di PT Suitmedia Kreasi Indonesia.

Tabel 3. 1. Linimasa Kerja Magang

Minggu	Jenis Pekerjaan yang Dilakukan
1	<p><i>Onboarding</i></p> <ul style="list-style-type: none"> • Pengenalan profile, <i>work culture</i>, acara dan struktur perusahaan. • Pengenalan divisi frontend, mentor, dan diskusi proyek percobaan. • Studi <i>boilerplate framework</i>, Nuxt JS, plugins, git, gitlab frontend Suitmedia
2	<ul style="list-style-type: none"> • Duplikasi halaman dokumentasi <i>boilerplate framework</i> frontend Suitmedia • Membuat proyek percobaan website <i>e-commerce</i> • User Interface Development • Presentasi dan diskusi perkembangan pengerjaan proyek

3	<ul style="list-style-type: none"> • Studi fetch API dan <i>plugins</i> Axios
---	--

Tabel 3. 2. Linimasa Kerja Magang (lanjutan)

Minggu	Jenis Kegiatan yang Dilakukan
3	<ul style="list-style-type: none"> • Briefing pengembangan pengembangan proyek PT Ritel dan Multi Manufaktur • Studi sitemap, desain UI, dan review code yang telah dibuat sebelumnya. • UI Development • Debug dan adjustment UI
4	<ul style="list-style-type: none"> • Bug fixing dan adjutment UI • Membuat halaman dinamis • Studi <i>best practice</i> integrasi API Nuxt JS • Integrasi data API
5	<ul style="list-style-type: none"> • Daily Meeting • Mengikuti <i>User Acceptance Testing</i> (UAT) • Debug issue dan adjustment dari UAT Sprint 3 • Debug dan isu UAT sprint 3 • Weekly meeting
6	<ul style="list-style-type: none"> • Daily Meeting • UI development sprint 4 • Integrasi API • Debug isu dan adjustment UAT sprint 3 • Weekly meeting
7	<ul style="list-style-type: none"> • Daily Meeting • Membuat fetch API error handling

Tabel 3. 3. Linimasa Kerja Magang (lanjutan)

Minggu	Jenis Kegiatan yang Dilakukan
7	<ul style="list-style-type: none"> • Integrasi API sprint 4 • Weekly meeting
8	<ul style="list-style-type: none"> • Daily meeting • Integrasi API landing page • Debug isu dan adjustment sprint 4 • Weekly meeting
9	<ul style="list-style-type: none"> • Daily meeting • Integrasi API • UI Adjustment • Weekly meeting
10	<ul style="list-style-type: none"> • Daily meeting • UI Development dan integrasi API registrasi acara • Debug isu SIT sprint 4 • Weekly meeting
11	<ul style="list-style-type: none"> • Daily meeting • UAT sprint 4 • Debug isu, dan adjustment UAT sprint 4 • UI Development sprint 5 • Weekly meeting
12	<ul style="list-style-type: none"> • Daily meeting • UI Development sprint 5 • API Integration sprint 5 • Debug isu SIT sprint 5

Tabel 3. 4. Linimasa Kerja Magang (lanjutan)

Minggu	Jenis Kegiatan yang Dilakukan
12	<ul style="list-style-type: none"> • Weekly meeting • Debug dan adjustment UAT sprint 4
13	<ul style="list-style-type: none"> • Daily meeting • Integrasi API sprint 5 • Debug dan adjustment UAT sprint 4 • Weekly meeting
14	<ul style="list-style-type: none"> • Daily meeting • Studi peningkatan performa web • Integrasi API sprint 5 • Debug SIT sprint 5 • Weekly meeting
15	<ul style="list-style-type: none"> • Daily meeting • Debug UAT sprint 4 • Integrasi API sprint 5 • Adjustment error handling • Diskusi perubahan alur fitur sistem. • Weekly meeting
16	<ul style="list-style-type: none"> • Daily meeting • Debug isu sprint 5 • UI Development dan adjustment sprint 5 • Weekly meeting
17	<ul style="list-style-type: none"> • Daily meeting • Diskusi perubahan salah satu alur fitur dengan backend dan analis

Tabel 3. 5. Linimasa Kerja Magang (lanjutan)

Minggu	Jenis Kegiatan yang Dilakukan
17	<ul style="list-style-type: none"> • Optimasi code data integrasi • Debug isu UAT sprint 5 • Adjustment sprint 5 • Weekly meeting
18	<ul style="list-style-type: none"> • Daily meeting • UI Development sprint 5 & sebagian sprint 6 • Debug isu sprint 5 • End to End testing • Review code • Studi optimasi performa • Weekly meeting
19	<ul style="list-style-type: none"> • Daily meeting • Kerja onsite Bandung bersama klien • Go live platform loyalitas pelanggan PT Ritel dan Multi Manufaktur • Debug isu live • Debug isu pelanggan
20	<ul style="list-style-type: none"> • Audit performa website • Review code • Optimasi performa website • Debug isu live dan isu dari pelanggan • Studi dan percobaan peningkatan performa dengan plugins baru
21	<ul style="list-style-type: none"> • Daily meeting • Peningkatan performa website • Debug isu live dan isu dari pelanggan

Tabel 3. 6. Linimasa Kerja Magang (lanjutan)

Minggu	Jenis Kegiatan yang Dilakukan
21	<ul style="list-style-type: none"> • Weekly meeting
22	<ul style="list-style-type: none"> • Daily meeting • Peningkatan performa website • Debug isu after live dan isu dari pelanggan • UI development dan integrasi API sprint 6 • Weekly meeting
23	<ul style="list-style-type: none"> • Daily meeting • Peningkatan performa website • Debug isu after live dan isu dari pelanggan • UI development, adjustment dan integrasi API sprint 6 • Weekly meeting
24	<ul style="list-style-type: none"> • Offboarding kerja magang

3.2.2. Uraian Kerja Magang

Program loyalitas pelanggan dimaksudkan untuk menarik atensi pelanggan dengan memberikan keuntungan kepada pelanggan PT Ritel dan Multi Manufaktur sehingga pelanggan tertarik untuk terus berbelanja dan tidak beralih ke produk kompetitor. Keuntungan yang pelanggan dapatkan dari program loyalitas adalah diberikannya poin atau kupon diskon ketika melakukan transaksi produk melalui *brandsite*, *e-commerce*, atau gerai resmi PT Ritel dan Multi Manufaktur. Fitur yang disediakan dalam platform loyalitas pelanggan adalah sebagai berikut.

- 1) Login, registrasi, dan reaktivasi akun dari platform yang sebelumnya telah dikembangkan oleh PT Ritel dan Multi Manufaktur.
- 2) Informasi promosi, acara, dan artikel dari oleh PT Ritel dan Multi Manufaktur.
- 3) Catatan histori transaksi pelanggan. Histori ini diintegrasikan dengan seluruh toko resmi dari PT Ritel dan Multi Manufaktur baik toko dalam *e-commerce* seperti Tokopedia, Shopee, Lazada, Brandsite, maupun gerai resmi yang tersebar di berbagai daerah Indonesia. Sehingga dimanapun pelanggan melakukan transaksi, pelanggan akan mendapatkan keuntungan dengan menggunakan platform tersebut.
- 4) Penukaran kupon diskon dan poin. Dalam platform loyalitas, pelanggan dapat membeli kupon diskon dengan poin yang didapatkan dari transaksi atau menyelesaikan misi, survey, membagikan informasi dan referal dalam platform tersebut. Poin yang didapatkan juga dapat ditukarkan untuk pembayaran ketika pelanggan melakukan transaksi pada gerai resmi PT Ritel dan Multi Manufaktur.
- 5) *Campaign* untuk pelanggan. *Campaign* yang dibuat dalam platform ini berupa misi atau survey. Dengan menyelesaikan *campaign* yang diberikan, pelanggan akan mendapatkan hadiah berupa poin atau kupon diskon.
- 6) Pencarian lokasi gerai. Pada platform ini pelanggan dapat mencari lokasi gerai resmi PT Ritel dan Multi Manufaktur. Lokasi terdekat akan didapatkan apabila pelanggan mengaktifkan GPS pada saat mengakses platform tersebut. Disediakan juga fitur untuk membuka google maps apabila pelanggan membutuhkan panduan menuju gerai.

Adapun tahapan yang dilakukan penulis selama pengembangan program loyalitas pelanggan adalah sebagai berikut.

A. Pengembangan Antarmuka Pengguna (User Interface Development)

Melalui platform Figma, hasil desain diberikan kepada divisi frontend dari divisi UI/UX. Yang selanjutnya desain tersebut diimplementasikan ke dalam code frontend. Dalam tahapan ini penulis mempelajari desain yang telah dibuat untuk menentukan kebutuhan pengembangan seperti *font*, *icon*, *plugins*, serta merancang penerapan terbaik dalam membuat komponen antarmuka.

Untuk antarmuka pengguna yang membutuhkan data baik untuk *state* antarmuka ataupun untuk mensimulasikan data yang tampil, penulis menggunakan *dummy data* yang disimpan dalam *framework*. *Dummy data* diperlukan untuk memastikan alur data dan *state* antarmuka berjalan sesuai dengan skema program yang telah dibuat. Implementasi desain antarmuka pengguna juga dibuat agar *responsive*. *Responsive* pada web mempunyai makna antarmuka pengguna mampu menyesuaikan tata letak tampilan dengan perangkat yang dipakai untuk mengaksesnya.

B. Integrasi Data dari Endpoint Backend dengan Sistem Frontend

Setelah pengembangan antarmuka pengguna selesai, selanjutnya dilakukan integrasi data dari sistem *backend* program loyalitas pelanggan. Tim backend telah menyediakan dokumentasi dengan menggunakan *tools* Swagger. Pada sisi *frontend*, *fetching* data menggunakan tambahan *plugins* Axios untuk memudahkan integrasi dan penanganan ketika terjadi *error* dalam prosesnya. Penanganan diperlukan untuk memberikan antisipasi sistem crash,

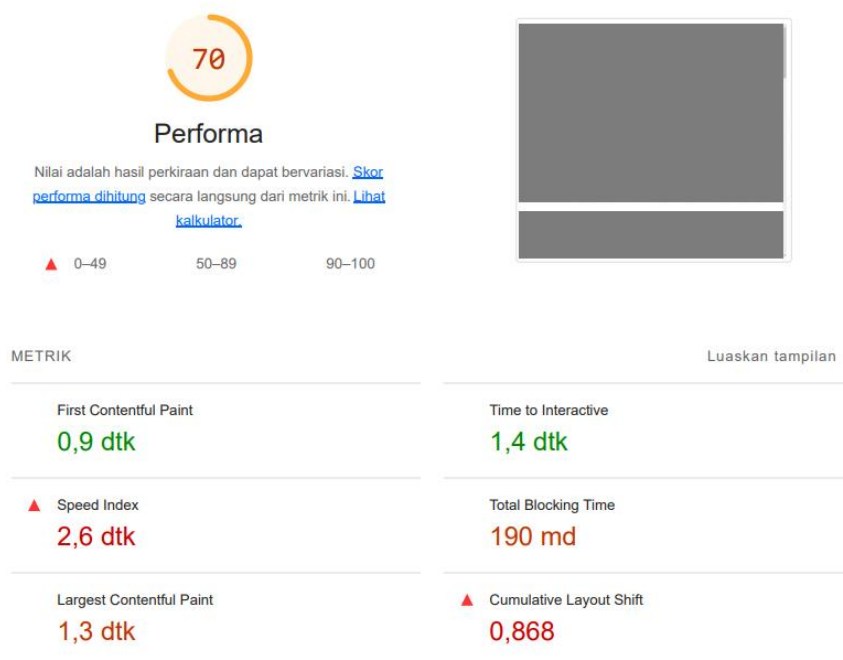
penyampaian error, percobaan ulang fetch, atau langkah yang perlu pelanggan lakukan ketika terjadi gagal fetch.

C. Adjustment dan Bug Fixing Issue dari System Integration Testing dan User Acceptance Testing

System Integration Testing (SIT) adalah tes yang dilakukan oleh tim analis sedangkan *User Acceptance Testing (UAT)* dilakukan bersama dengan klien, yaitu PT Ritel dan Multi Manufaktur. Dari tes tersebut dapat ditemukan *bug* atau *issue* pada sistem yang dikembangkan. Kemudian dilakukan perbaikan bug yang ditemukan dan berkoordinasi dengan tim backend apabila *bug* yang ditemukan berkaitan dengan data dari *API endpoint*.

D. Peningkatan Performa Sistem

Analisa dan audit performa pada sisi *frontend* dilakukan dengan fitur Lighthouse yang terdapat pada menu browser Chrome. Lighthouse melakukan penilaian performa dengan enam parameter yaitu *First Contentful Paint (FCP)*, *speed index*, *Time to Interactive (TTI)*, *Total Blocking Time (TBT)*, *Largest Contentful Paint (LCP)*, dan *Cumulative Layout Shift (CLS)*. Berikut adalah hasil dari audit lighthouse *desktop* untuk satu sampel halaman pada website platform program loyalitas pelanggan sebelum dilakukan optimasi.



Gambar 3. 2. Sampel Tes Performa Halaman Web Platform Loyalitas Pelanggan

First Contentfull Paint (FCP) adalah parameter yang mengukur waktu saat halaman web mulai dimuat hingga munculnya satu konten elemen yang tampil di layar. Peningkatan nilai FCP dapat dilakukan dengan mengeliminasi *resources* yang menahan proses *rendering*. Penulis melakukan peningkatan dengan mengeliminasi *styling class* (CSS), mengoptimalkan penggunaan *function*, dan *component* pada *framework* untuk mengurangi ukuran *payload* dan waktu *parsing script* saat browser melakukan *request* halaman web.

Largest Contentfull Paint (LCP) merepresentasikan seberapa cepat konten utama dalam suatu web ditampilkan sesaat setelah pengguna menginisiasi request untuk mengakses halaman web. Peningkatan *LCP* juga berkaitan dengan *FCP*, dimana ukuran dari konten yang dimuat sangat berpengaruh. Selain mengoptimalkan *code* untuk mengurangi *payload* dan waktu

parsing script, melakukan kompresi konten gambar, *icon*, dan aset lainnya juga dapat meningkatkan nilai performa *LCP*.

Ketika browser melakukan permintaan ke server, server akan mengirimkan *bundle* data yang diperlukan untuk menampilkan halaman yang diminta. Jika ukuran *bundle* terlalu besar akan menyebabkan block time untuk browser melakukan render. Karena itu diperlukan *code splitting* untuk membagi satu *bundle* yang besar menjadi beberapa bagian bundle yang berukuran lebih kecil atau disebut *chunk*. Konfigurasi yang digunakan pada *Nuxt JS* untuk optimasi performa dengan menerapkan menerapkan *split code* ditunjukkan pada gambar 3.3.

```
build: {
  extend(config, ctx) { },
  postcss: {
    preset: {
      autoprefixer: {
        overrideBrowserslist: ['last 2 versions']
      }
    }
  },
  optimizecss: true,
  splitChunks: {
    pages: true,
    vendor: true,
    commons: true,
    runtime: true,
    layouts: true
  },
  optimization: {
    splitChunks: {
      minSize: 20000,
      maxSize: 500000
    }
  }
}
```

Gambar 3. 3. Snippet Code Konfigurasi Build Nuxt JS

Format gambar yang digunakan dalam web juga berpengaruh terhadap kompresi ukuran gambar. Digunakan format gambar webp untuk mengoptimasi *LCP* pada web program loyalitas pelanggan PT Ritel dan Manufaktur. Namun tidak semua browser sudah mendukung penggunaan format gambar webp. Karena itu

diperlukan solusi untuk membuat browser melakukan *fallback* ke format gambar non webp apabila browser tidak mendukung format webp. Untuk menyelesaikan permasalahan tersebut diperlukan *middleware* untuk mengkonversi format image ke dalam webp. Namun pada Nuxt JS, *library* Nuxt-image yang mendukung konversi format webp hanya dapat digunakan untuk mode *server side rendering*. Sedangkan platform yang dibangun berbasis *client side rendering*. Untuk itu dilakukan pendekatan dengan menggunakan tag `<picture>` untuk membuat *fallback image* apabila browser tidak mendukung representasi data webp.

```
<picture v-lazy-load>
  <source data-srcset="/assets/img/value1.webp" type="image/webp" />
  
</picture>
```

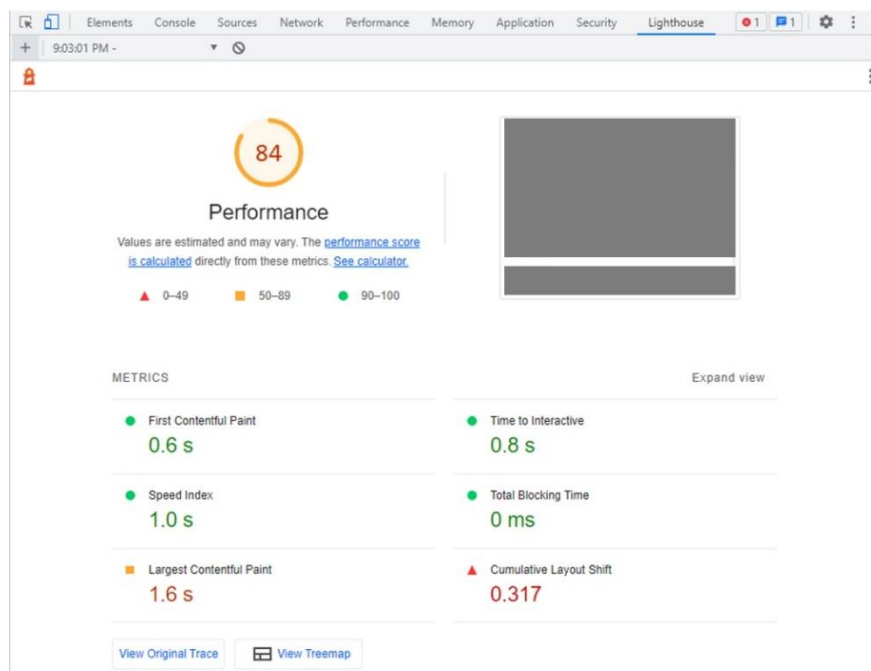
Gambar 3. 4. Snippet Code Penggunaan Tag Picture pada HTML

Merujuk *code* Gambar 3.4, browser yang sudah mendukung representasi data webp akan menampilkan *value1.webp*. Browser yang tidak mendukung akan melakukan *fallback* ke *source value1.png*. Dan pada browser versi lama seperti *Internet Explorer* yang tidak mendukung tag `<picture>` akan *fallback* ke tag ``.

Parameter pengukuran performa selanjutnya adalah *Total Blocking Time (TBT)*. TBT merupakan parameter yang mengukur jeda waktu antara saat halaman dimuat (*FCP*) hingga saat *sub-resources* dimuat dan mampu untuk merespon inputan dari pengguna (*TTI*). Nilai *TBT* dapat ditingkatkan dengan mengurangi jumlah permintaan *resources* yang diperlukan untuk memuat keseluruhan tampilan web. Alternatif ketika *content request* tidak bisa dikurangi adalah dengan menunda *request* konten untuk dimuat secara bersamaan. Seperti menggunakan *lazy load* pada gambar yang tidak muncul pada *viewport* pengguna saat memuat halaman web.

Nilai *Cumulative Layout Shift (CLS)* didapatkan dari besarnya pergeseran tata letak konten web yang tidak terduga selama halaman diakses. Nilai *CLS* pada platform loyalitas pelanggan ini berada di atas 0.2 dikarenakan sebagian besar konten halaman dirender berdasarkan *state* data dari *endpoint* yang diambil secara *asynchronus*. Ketika proses pengambilan data selesai terjadi perubahan data lalu browser akan memperbarui *DOM (Document Object Model)* sehingga terjadi *layout shifting* atau pergeseran tata letak konten. Untuk menurunkan nilai *CLS*, digunakan *styling CSS* pada elemen untuk mengalokasikan tempat untuk konten yang akan muncul setelah proses pengambilan data selesai.

Setelah dilakukan optimasi performa untuk halaman sampel website platform program loyalitas pelanggan didapatkan nilai audit lighthouse sebesar 84.



Gambar 3. 5. Hasil Audit Lighthouse Setelah Dilakukan Optimasi

Untuk keseluruhan jumlah halaman pada website ini adalah 35. Kemudian dilakukan audit untuk setiap halamannya dan

didapatkan nilai rata-rata untuk keseluruhan halaman adalah 85.41 dengan nilai tertinggi 98 dan terendah 71.

E. Go Live

Go live adalah istilah yang digunakan untuk proses peralihan dari *development environment* ke *production*. Pada tahap ini penulis bekerja sama dengan seluruh tim dan klien untuk memastikan keseluruhan sistem yang telah dibuat bekerja dengan baik. Pada tahap ini layanan *helpdesk* dibuka oleh PT Ritel dan Multi Manufaktur untuk pelanggan yang mengalami masalah dalam menggunakan platform program loyalitas pelanggan. Penulis dan tim berkoordinasi untuk menyelesaikan setiap masalah yang dilaporkan ke *helpdesk* secara cepat dan hati-hati agar proses *fixing* tidak mengganggu proses bisnis PT Ritel dan Multi Manufaktur yang sedang berjalan.

3.2.3. Kendala yang Ditemukan

Kendala yang dihadapi selama pembuatan platform program loyalitas pelanggan PT Ritel dan Multi Manufaktur adalah kurangnya wawasan dan pengetahuan penulis tentang *environment*, logika, dan *best practices* penggunaan *framework* Nuxt JS. Ketidaktahuan penulis seringkali menjadi penyebab lambatnya penyelesaian suatu fitur yang dikerjakan.

3.2.4. Solusi atas Kendala yang Ditemukan

Untuk menghadapi kendala yang dialami, penulis mencari referensi terkait Nuxt JS baik yang disediakan oleh PT Suitmedia Kreasi Indonesia maupun hasil mencari referensi pribadi. Penulis juga melakukan diskusi dengan supervisi, analis, tim frontend, backend, dan Project Manager dari klien apabila terdapat *issue* atau *bug* dalam program untuk mendapatkan solusi yang tepat.