

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Kerja Praktek

Hasil pekerjaan yang sudah dilaksanakan yaitu membuat Sistem Informasi Dapur milik Divisi Dapur tersedia untuk perangkat smartphone Android menggunakan teknologi *Cloud Computing* web untuk mempermudah Divisi Dapur YSI menerima ringkasan data stok dapur, kunjungan, dan pemantauan inventaris.

Setelah perangkat lunak diuji coba fungsionalitasnya dan berjalan baik, maka perangkat lunak mulai digunakan Divisi Dapur YSI dengan pendampingan Divisi IT YSI

4.2 Deskripsi Perangkat Lunak

Tampilan perangkat lunak menggunakan konsep dari material design google sehingga pengguna sudah terbiasa dan mudah beradaptasi dengan program. Pengembangan Perangkat lunak menggunakan konsep Object Oriented sehingga pembahasan desain dan perencanaan menggunakan UML.

4.2.1 Modul Program

1. Bagian Login

Pada bagian ini menampilkan fitur untuk login ke dalam perangkat lunak, modulnya adalah LoginActivity, LoginAPI dan LoginData.

```
public class LoginActivity extends AppCompatActivity {
    private String email, password; private TextView Email, Password;
    private KontrolData kontrolData;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        SharedPreferences pref = getSharedPreferences("akun", MODE_PRIVATE);
        kontrolData = new KontrolData(); kontrolData.setKontrol(pref);
        kontrolData.readProfil(); loginStat();
        Email = findViewById(R.id.login_edit_email);
        Password = findViewById(R.id.login_edit_password);
        Button login = findViewById(R.id.login_button_login);
        Email.setText(kontrolData.getEmail())
        login.setOnClickListener(new View.OnClickListener() {
```

```

        @Override
        public void onClick(View v) {
            email = Email.getText().toString();
            password = Password.getText().toString();
            LoginAPI();}});}
    private void LoginAPI(){
        final LoadingDialog loadingDialog = new
LoadingDialog(LoginActivity.this);
        final RuntimeDialog runtimeDialog = new RuntimeDialog();
        loadingDialog.setPesan(getString(R.string.dialog_login_akun));
        loadingDialog.show();
        runtimeDialog.setTitle(getString(R.string.dialog_login_fail_title));
        runtimeDialog.setMsg(getString(R.string.dialog_login_fail_msg));
        Service service = Client.getClient().create(Service.class);
        Call<LoginAPI> call = service.login(email, password);
        call.enqueue(new Callback<LoginAPI>() {
            @Override
            public void onResponse(Call<LoginAPI> call, Response<LoginAPI>
response) {
                if(response.body() != null){
                    String status = response.body().getStatus(),
                        API_KEY = response.body().getTokenAccess();
                    loadingDialog.dismiss();
                    LoginData Data = response.body().getData();
                    if(status.equals("success")){
                        kontrolData.setEmail(email);
                        kontrolData.setPassword(password);
                        kontrolData.setApi_key(API_KEY);
                        kontrolData.setState(1);
                        kontrolData.setNama(Data.getName());
                        kontrolData.setRole_id(Data.getRoleId());
                        kontrolData.setUser_id(Data.getId());
                        kontrolData.writeProfil();
                        Intent goToMain = new Intent(getApplicationContext(),
MainActivity.class);
                        goToMain.putExtra("API_KEY", API_KEY);
                        startActivity(goToMain);
                        finishAffinity();
                    }else{runtimeDialog.show(getSupportFragmentManager(), "");
                        }}}
            @Override
            public void onFailure(Call<LoginAPI> call, Throwable t) {
                runtimeDialog.setMsg(getString(R.string.dialog_load_fail_msg));
                runtimeDialog.show(getSupportFragmentManager(), "");
                loadingDialog.dismiss();
            }
        });
    }
    private void loginStat(){
        if(kontrolData.getState() == 1){
            email = kontrolData.getEmail();
            password = kontrolData.getPassword(); LoginAPI();
        }
    }
}

```

Modul Program 4.2.1 LoginActivity

```

public class LoginAPI {
    @SerializedName("status")@Expose private String status;
    @SerializedName("token_access") @Expose private String tokenAccess;
    @SerializedName("data") @Expose private LoginData data;
    public String getStatus() { return status; }
    public void setStatus(String status) { this.status = status; }
    public String getTokenAccess() { return tokenAccess; }
    public void setTokenAccess(String tokenAccess) {
        this.tokenAccess = tokenAccess; }
    public LoginData getData() { return data; }
    public void setData(LoginData data) { this.data = data; }
}

```

Modul Program 4.2.2 LoginAPI

```

public class LoginData {
    @SerializedName("id")@Exposeprivate String id;
    @SerializedName("name")@Expose private String name;
    @SerializedName("role_id")@Expose private String roleId;
    public String getId() { return id; }
    public void setId(String id) { this.id = id; }
    public String getName() { return name; }
    public void setName(String name) { this.name = name; }
    public String getRoleId() { return roleId; }
    public void setRoleId(String roleId) { this.roleId = roleId; }
}

```

Modul Program 4.2.3 LoginData

4:56 AM XL 4G 43%

TELADAN
KB•TK•SD YOGYAKARTA
BERAKHLAK • QURANI • SMART

DIVISI DAPUR

Alamat Email
admin@teladan.com

Kata Sandi

LOGIN

INFORMASI :
Lupa Password? Hubungi Staf IT Sekolah Teladan

Gambar 4.2.1 Tampilan login

2. Bagian MainActivity

Bagian ini merupakan controller dari untuk navigasi pengguna ke berbagai menu lain.

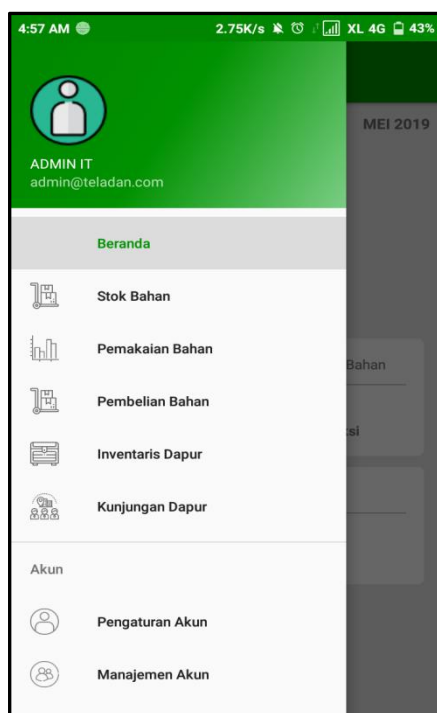
```
public class MainActivity extends AppCompatActivity implements
NavigationView.OnNavigationItemSelectedListener {
    private FragmentManager fragmentManager = getSupportFragmentManager();
    private TextView Nama, Email; private SharedPreferences pref;
    private KontrolData kontrolData; private Menu nav_manajemen;
    private LoaderAPI loaderAPI = new LoaderAPI();
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        NavigationView navigationView = findViewById(R.id.nav_view);
        nav_manajemen = navigationView.getMenu();
        navigationView.setNavigationItemSelectedListener(this);
        navigationView.setCheckedItem(R.id.nav_beranda);
        View nav_header = navigationView.getHeaderView(0);
        Nama = nav_header.findViewById(R.id.nav_header_nama);
        Email = nav_header.findViewById(R.id.nav_header_email);
        pref = getSharedPreferences("akun", MODE_PRIVATE);
        kontrolData = new KontrolData(); initiate();
        fragmentManager.beginTransaction().replace(R.id.content_main, new
Beranda()).commit();
        private void initiate() {kontrolData.setKontrol(pref);
kontrolData.readProfil();>Nama.setText(kontrolData.getNama());
        Email.setText(kontrolData.getEmail());
        if (!"2".equals(kontrolData.getRole_id())
        && !"3".equals(kontrolData.getRole_id()))
        {nav_manajemen.findItem(R.id.nav_akun_manajemen).setVisible(false);}
        loaderAPI.prepare(MainActivity.this, getSupportFragmentManager(),
kontrolData);}
        @Override
        public void onBackPressed() {
            DrawerLayout drawer = findViewById(R.id.drawer_layout);
            if (drawer.isDrawerOpen(GravityCompat.START)) {
                drawer.closeDrawer(GravityCompat.START);
            } else { super.onBackPressed(); }
        }
        @Override
        public boolean onNavigationItemSelected(@NonNull MenuItem item) {
            int id = item.getItemId(),
                frag_stack = fragmentManager.getBackStackEntryCount();
            if (frag_stack > 0) { fragmentManager.popBackStack(); }
            switch (id) {
                case R.id.nav_beranda:
                    fragmentManager.beginTransaction().replace(R.id.content_main, new
Beranda()).commit(); break;
                case R.id.nav_stockbahan:
                    fragmentManager.beginTransaction().replace(R.id.content_main, new
StokActivity()).commit(); break;
                case R.id.nav_pemakaian: loaderAPI.load(id); break;
                case R.id.nav_pembelian: loaderAPI.load(id); break;
                case R.id.nav_inventaris:
                    fragmentManager.beginTransaction().replace(R.id.content_main,
```

```

new InventarisActivity()).commit(); break;
    case R.id.nav_kunjungan: loaderAPI.load(id); break;
    case R.id.nav_akun:
        fragmentManager.beginTransaction().replace(R.id.content_main, new
        AkunProfilku()).commit(); break;
    case R.id.nav_akun_manajemen:
        fragmentManager.beginTransaction().replace(R.id.content_main, new
        ManajemenList()).commit(); break;}
    DrawerLayout drawer = findViewById(R.id.drawer_layout);
    drawer.closeDrawer(GravityCompat.START); return true;}}

```

Modul Program 4.2.4 MainActivity



Gambar 4.2.2 Tampilan Navigasi Perangkat Lunak

3. Bagian Ringkasan Laporan (Beranda)

Bagian ini menampilkan fitur untuk melihat ringkasan laporan dari penggunaan dan pembelian bahan dapur, modulnya adalah Beranda, RingkasanAPI dan Ringkasan.

```

public class Beranda extends Fragment {
    private AppCompatActivity activity; private KontrolData kontrolData;
    private TextView Beli, Pakai, Harga; public Beranda() {}
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, ViewGroup
    container, Bundle savedInstanceState) {
        TextView user, bulan; activity = (AppCompatActivity) getActivity();
        View view = inflater.inflate(R.layout.fragment_beranda, container,
        false); Toolbar toolbar = view.findViewById(R.id.toolbar);
        Beli = view.findViewById(R.id.card_beranda_pembelian_bahan);
        Pakai = view.findViewById(R.id.card_beranda_pemakaian_bahan);
        Harga = view.findViewById(R.id.card_beranda_total_pengeluaran);
    }
}

```

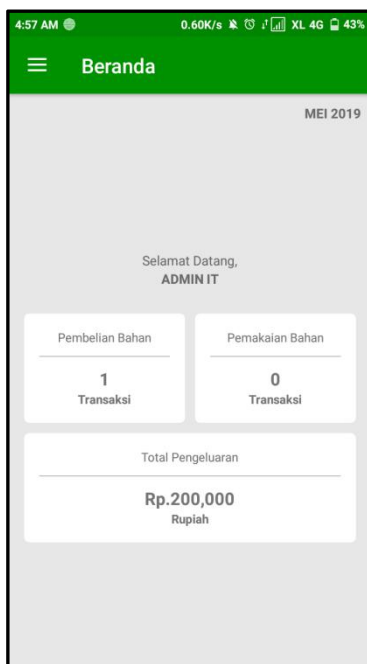
```

        user = view.findViewById(R.id.card_beranda_info_user);
        bulan = view.findViewById(R.id.beranda_bulan);
        activity.supportActionBar(toolbar); activity.setTitle("Beranda");
        DrawerLayout drawer = activity.findViewById(R.id.drawer_layout);
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(activity,
        drawer, toolbar, R.string.navigation_drawer_open,
        R.string.navigation_drawer_close);
        drawer.addDrawerListener(toggle); toggle.syncState();
        SharedPreferences pref = activity.getSharedPreferences("akun",
        MODE_PRIVATE);
        kontrolData = new KontrolData(); kontrolData.setKontrol(pref);
        kontrolData.readProfil(); user.setText(kontrolData.getNama());
        bulan.setText(kontrolData.getFormat_BulanSekarang().toUpperCase(Locale.ge
        tDefault())); ringkasanAPI(); return view;
    }
    private void ringkasanAPI() {
        final LoadingDialog loadingDialog = new LoadingDialog(getContext());
        final RuntimeDialog runtimeDialog = new RuntimeDialog();
        loadingDialog.setPesan(getString(R.string.dialog_load_data));
        loadingDialog.show();
        runtimeDialog.setTitle(getString(R.string.dialog_load_fail_title));
        runtimeDialog.setMsg(getString(R.string.dialog_load_fail_msg));
        Service service = Client.getClient().create(Service.class);
        Call<RingkasanAPI> call =
        service.getRingkasan(kontrolData.getBulan_sekarang(), getAPIKEY());
        call.enqueue(new Callback<RingkasanAPI>() {
            @Override
            public void onResponse(Call<RingkasanAPI> call,
            Response<RingkasanAPI> response) {
                if(response.body() != null){
                    String status = response.body().getStatus();
                    loadingDialog.dismiss();
                    if (status.equals("success")) {
                        Ringkasan ringkasan = response.body().getRingkasan();

                        Beli.setText(ringkasan.getMasuk()); Pakai.setText(ringkasan.getPakai());
                        String harga = "Rp."+ringkasan.getHarga(); Harga.setText(harga);}}else {
                            runtimeDialog.show(getChildFragmentManager(), "");}}
            @Override
            public void onFailure(Call<RingkasanAPI> call, Throwable t) {
                Toast.makeText(getContext(), "Failed to load",
                Toast.LENGTH_LONG).show();
                loadingDialog.dismiss();
                runtimeDialog.setMsg(t.getMessage());
                runtimeDialog.show(getChildFragmentManager(), "");
            }
        });
    }
    private String getAPIKEY() {return kontrolData.getApi_key();}

```

Modul Program 4.2.5 Beranda



Gambar 4.2.3 Tampilan Ringkasan Laporan (Beranda)

4. Bagian Laporan Stok Bahan

Bagian ini menampilkan fitur untuk melihat laporan stok bahan, yaitu stok bahan dapur yang ada pada hari berjalan, modulnya adalah StokActivity, StokBahanAPI dan StokBahan.

```
public class StokActivity extends Fragment {
    private List<StokBahan> stokBahanList = new ArrayList<>();
    private SharedPreferences pref; private KontrolData kontrolData;
    RecyclerView recyclerView;
    public StokActivity() { }
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, ViewGroup
    container,
                           Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_stok_bahan,
    container, false);
        Toolbar toolbar = view.findViewById(R.id.toolbar);
        AppCompatActivity activity = (AppCompatActivity) getActivity();
        recyclerView = view.findViewById(R.id.stok_bahan_list_item);
        activity.supportActionBar(toolbar);
        activity.setTitle("Stok Bahan Dapur");
        DrawerLayout drawer = activity.findViewById(R.id.drawer_layout);
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
            activity, drawer, toolbar, R.string.navigation_drawer_open,
    R.string.navigation_drawer_close);
        drawer.addDrawerListener(toggle); toggle.syncState();
        pref = activity.getSharedPreferences("akun", MODE_PRIVATE);
        kontrolData = new KontrolData(); loaderAPI();
        RecyclerView.LayoutManager mLayoutManager = new
```

```

LinearLayoutManager(getContext());
recyclerView.setLayoutManager(mLayoutManager);
recyclerView.setItemAnimator(new DefaultItemAnimator());return view;}
private void loaderAPI(){
    final LoadingDialog loadingDialog = new LoadingDialog(getContext());
    final RuntimeDialog runtimeDialog = new RuntimeDialog();
    loadingDialog.setPesana(getString(R.string.dialog_load_data));
    loadingDialog.show();
    runtimeDialog.setTitle(getString(R.string.dialog_load_fail_title));
    runtimeDialog.setMsg(getString(R.string.dialog_load_fail_msg));
    Service service = Client.getClient().create(Service.class);
    Call<StokBahanAPI> call = service.getStokBahan(getAPIKEY());
    call.enqueue(new Callback<StokBahanAPI>() {@Override
        public void onResponse(Call<StokBahanAPI> call,
Response<StokBahanAPI> response) {
if(response.body() != null){String status = response.body().getStatus();
loadingDialog.dismiss();if(status.equals("success")){
stokBahanList = response.body().getStokBahan();
StokListAdapter stokListAdapter = new StokListAdapter(stokBahanList);
recyclerView.setAdapter(stokListAdapter);}}else{
        runtimeDialog.show(getChildFragmentManager(), "");
        }}@Override
        public void onFailure(Call<StokBahanAPI> call, Throwable t) {
loadingDialog.dismiss();runtimeDialog.show(getChildFragmentManager(),
"");}}});
private String getAPIKEY(){kontrolData.setKontrol(pref);
    kontrolData.readProfil();return kontrolData.getApi_key();}}

```

Modul Program 4.2.6 StokActivity

```

public class StokBahanAPI{
    @SerializedName("status")@Exposeprivate String status;
    @SerializedName("stok_bahan")@Expose
    private List<StokBahan> stokBahan = null;
    public String getStatus() {return status;}
    public void setStatus(String status) {this.status = status;}
    public List<StokBahan> getStokBahan() {return stokBahan;}
    public void setStokBahan(List<StokBahan> stokBahan) {
        this.stokBahan = stokBahan;}
}

```

Modul Program 4.2.7 StokBahanAPI

```

public class StokBahan {
    @SerializedName("nm_stok")@Expose private String nmStok;
    @SerializedName("jumlah")@Expose private String jumlah;
    @SerializedName("nm_satuan")@Expose private String nmSatuan;
    public String getNmStok() { return nmStok; }
    public void setNmStok(String nmStok) { this.nmStok = nmStok; }
    public String getJumlah() { return jumlah; }
    public void setJumlah(String jumlah) { this.jumlah = jumlah; }
    public String getNmSatuan() { return nmSatuan; }
    public void setNmSatuan(String nmSatuan) { this.nmSatuan = nmSatuan; }
}

```

Modul Program 4.2.8 StokBahan

DAFTAR STOK BAHAN DAPUR	HARI INI
Bawang Merah 13708.2999999999927240423858165740966796875 Kilogram	
Bawang Putih 6997.5 Kilogram	
Bayam 0 Ikat	
Beras 7710.199999999998181010596454143524169921875 Kilogram	
Cabai 394 Gram	
Kangkung 0 Ikat	
Lengkuas 0 Ikat	
Minyak Goreng 107.5 Liter	
Telur 43626 Butir	

Gambar 4.2.4 Tampilan Laporan Stok Bahan

5. Bagian Laporan Pemakaian

Bagian ini menampilkan fitur untuk melihat laporan pemakaian bahan dapur modulnya adalah PemakaianActivity, PemakaianAdapter, PemakaianAPI, DataGrafik, DaftarTransaksi, dan BahanPakai.

```
public class PemakaianActivity extends Fragment {
    private AppCompatActivity activity; private SharedPreferences pref;
    private KontrolData kontrolData; private FragmentManager fragmentManager;
    public PemakaianActivity() {} @Override
    public View onCreateView(@NonNull LayoutInflater inflater, ViewGroup
    container, Bundle savedInstanceState) {
        activity = (AppCompatActivity) getActivity();
        View view = inflater.inflate(R.layout.fragment_view_pager_main,
        container, false);
        ViewPager vPager = view.findViewById(R.id.main_viewpager);
        Toolbar toolbar = view.findViewById(R.id.toolbar);

        if(activity!=null){
            activity.setSupportActionBar(toolbar); activity.
            setTitle("Pemakaian");
            DrawerLayout drawer =
            activity.findViewById(R.id.drawer_layout);
            ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
                activity, drawer, toolbar,
                R.string.navigation_drawer_open, R.string.navigation_drawer_close);
            drawer.addDrawerListener(toggle); toggle.syncState();
            fragmentManager = getFragmentManager();
            pref = activity.getSharedPreferences("akun", MODE_PRIVATE);
            PemakaianAdapter pemakaianAdapter = new
            PemakaianAdapter(getFragmentManager());
            vPager.setAdapter(pemakaianAdapter);
            TabLayout tLayout = view.findViewById(R.id.main_tab_layout);
```

```

        tLayout.setupWithViewPager(vPager);
        tLayout.setTabMode(TabLayout.MODE_FIXED);initiate();return view;}
private void initiate(){kontrolData.setKontrol(pref);
    kontrolData.readProfil();}@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);setHasOptionsMenu(true);}
@Overridepublic void onCreateOptionsMenu(Menu menu, MenuInflater
inflater) {inflater.inflate(R.menu.menu_pilih_bulan, menu);}
@Overridepublic boolean onOptionsItemSelected(MenuItem item) {
    if(item.getItemId() == R.id.menu_bulan){
        MonthSelectorDialog monthSelectorDialog = new
MonthSelectorDialog();
        monthSelectorDialog.prepare(fragmentManager,
R.id.nav_pemakaian, kontrolData);
        monthSelectorDialog.show(fragmentManager, "");
    }return true;}}

```

Modul Program 4.2.9 PemakaianActivity

```

public class PemakaianAdapter extends FragmentStatePagerAdapter {
    private static int PAGE_COUNT=2;
    private String[] Pemakaiadapter = {"GRAFIK PEMAKAIAN", "DETAIL
PEMAKAIAN"};
    public PemakaianAdapter(FragmentManager fm){super(fm);}
    @Override
    public Fragment getItem(int position){
        switch(position){
            case 0:return new GrafikPemakaian();
            case 1:return new DetailPemakaian();}return null;}
    @Override public int getCount(){return PAGE_COUNT;}
    @Override
    public CharSequence getPageTitle(int position){
        return Pemakaiadapter[position];}}

```

Modul Program 4.2.10 PemakaianAdapter

```

public class PemakaianAPI {
    @SerializedName("status")@Expose private String status;
    @SerializedName("transaksi") @Expose private Integer transaksi;
    @SerializedName("bahan_pakai") @Expose
    private List<BahanPakai> bahanPakai = null;
    public String getStatus() { return status;}
    public void setStatus(String status) { this.status = status; }
    public Integer getTransaksi() { return transaksi;}
    public void setTransaksi(Integer transaksi) {this.transaksi = transaksi;}
    public List<BahanPakai> getBahanPakai() {return bahanPakai;}
    public void setBahanPakai(List<BahanPakai> bahanPakai) {
        this.bahanPakai = bahanPakai;}
}

```

Modul Program 4.2.11 PemakaianAPI

```

public class DataGrafik {
    @SerializedName("id_master_stok")@Expose private String idMasterStok;
    @SerializedName("jumlah") @Expose private String jumlah;
    @SerializedName("nm_satuan") @Expose private String nmSatuan;
    @SerializedName("tanggal_pakai") @Expose private String tanggalPakai;
    public void setJumlah(String jumlah) { this.jumlah = jumlah; }
    public void setIdMasterStok(String idMasterStok) {
        this.idMasterStok = idMasterStok; }
    public void setNmSatuan(String nmSatuan) { this.nmSatuan = nmSatuan; }
}

```

```

public void setTanggalPakai(String tanggalPakai) {
    this.tanggalPakai = tanggalPakai;}
public String getIdMasterStok() {return idMasterStok;}
public String getJumlah() { return jumlah;}
public String getNmSatuan() {return nmSatuan;}
public String getTanggalPakai() {return tanggalPakai;}
}

```

Modul Program 4.2.12 DataGrafik

```

public class DaftarTransaksi {
    @SerializedName("id_master_stok")@Expose private String idMasterStok;
    @SerializedName("nm_stok") @Expose private String nmStok;
    @SerializedName("jumlah") @Expose private String jumlah;
    @SerializedName("nm_satuan") @Expose private String nmSatuan;
    @SerializedName("name") @Exposeprivate String name;
    @SerializedName("nm_menu")@Expose private String nm_menu;
    @SerializedName("tanggal_pakai")@Expose private String tanggalPakai;
    public String getIdMasterStok() { return idMasterStok; }
    public void setIdMasterStok(String idMasterStok) {this.idMasterStok =
idMasterStok; }
    public String getNmStok() { return nmStok;}
    public void setNmStok(String nmStok) { this.nmStok = nmStok;}
    public String getJumlah() { return jumlah; }
    public void setJumlah(String jumlah) { this.jumlah = jumlah; }
    public String getNmSatuan() { return nmSatuan; }
    public void setNmSatuan(String nmSatuan) { this.nmSatuan = nmSatuan;}
    public String getName() { return name; }
    public void setName(String name) { this.name = name; }
    public void setNm_menu(String nm_menu) { this.nm_menu = nm_menu; }public
String getNm_menu() { return nm_menu; }
    public String getTanggalPakai() { return tanggalPakai; }
    public void setTanggalPakai(String tanggalPakai) {this.tanggalPakai =
tanggalPakai;}}

```

Modul Program 4.2.13 DaftarTransaksi

```

public class BahanPakai {
    @SerializedName("id_master_stok")@Exposeprivate String idMasterStok;
    @SerializedName("transaksi") @Expose private String transaksi;
    @SerializedName("nm_stok") @Expose private String nmStok;
    @SerializedName("jumlah") @Expose private String jumlah;
    @SerializedName("nm_satuan") @Expose private String nmSatuan;
    @SerializedName("tanggal_pakai") @Expose private String tanggalPakai;
    @SerializedName("grafik") @Expose
private List<DataGrafik> dataGrafik = null;
    @SerializedName("daftar_transaksi") @Expose
private List<DaftarTransaksi> daftarTransaksi = null;
    public String getIdMasterStok() { return idMasterStok; }
    public void setIdMasterStok(String idMasterStok) {
        this.idMasterStok = idMasterStok;}
    public String getTransaksi() { return transaksi;}
    public void setTransaksi(String transaksi) { this.transaksi = transaksi;}
    public String getNmStok() {return nmStok;}
    public void setNmStok(String nmStok) {this.nmStok = nmStok;}
    public String getJumlah() {return jumlah;}
    public void setJumlah(String jumlah) {this.jumlah = jumlah;}
    public String getNmSatuan() {return nmSatuan;}
    public void setNmSatuan(String nmSatuan) {this.nmSatuan = nmSatuan;}
    public String getTanggalPakai() {return tanggalPakai;}
}

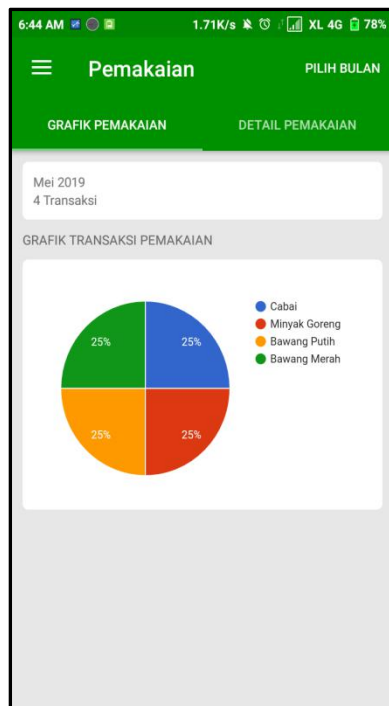
```

```

public void setTanggalPakai(String tanggalPakai) {
    this.tanggalPakai = tanggalPakai;
}
public List<DaftarTransaksi> getDaftarTransaksi() {
    return daftarTransaksi;
}
public List<DataGrafik> getDataGrafik() {return dataGrafik;}
public void setDataGrafik(List<DataGrafik> dataGrafik) {
    this.dataGrafik = dataGrafik;
}
public void setDaftarTransaksi(List<DaftarTransaksi> daftarTransaksi) {
    this.daftarTransaksi = daftarTransaksi;
}

```

Modul Program 4.2.14 BahanPakai



Gambar 4.2.5 Tampilan Grafik Pemakaian

Bahan	Jumlah	Transaksi
Cabai	200.00 Gram	1
Minyak Goreng	0.80 Liter	1
Bawang Putih	0.40 Kilogram	1
Bawang Merah	0.50 Kilogram	1

Gambar 4.2.6 Tampilan Detail Pemakaian

6. Bagian Laporan Pembelian

Bagian ini menampilkan fitur untuk melihat laporan pembelian bahan dapur modulnya adalah PembelianActivity, PembelianAdapter, PembelianAPI, DaftarTransaksi, BahanMasuk, dan Ringkasan.

```

public class PembelianActivity extends Fragment {
    private AppCompatActivity activity; private SharedPreferences pref;
    private KontrolData kontrolData; private FragmentManager fragmentManager;
    public PembelianActivity() { } @Override
    public View onCreateView(@NonNull LayoutInflater inflater, ViewGroup
    container, Bundle savedInstanceState) {
        activity = (AppCompatActivity) getActivity();
        View view = inflater.inflate(R.layout.fragment_view_pager_main,
        container, false);
        ViewPager vPager = view.findViewById(R.id.main_viewpager);
        Toolbar toolbar = view.findViewById(R.id.toolbar);
        activity.supportActionBar(toolbar); activity.setTitle("Pembelian");
    }
}

```

```

        fragmentManager = getFragmentManager();
        pref = activity.getSharedPreferences("akun", MODE_PRIVATE);
        kontrolData = new KontrolData();
        DrawerLayout drawer = activity.findViewById(R.id.drawer_layout);
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
            activity, drawer, toolbar, R.string.navigation_drawer_open,
            R.string.navigation_drawer_close);
        drawer.addDrawerListener(toggle); toggle.syncState();
        PembelianAdapter pembelianAdapter = new
        PembelianAdapter(getFragmentManager());
        viewPager.setAdapter(pembelianAdapter);
        TabLayout tLayout = view.findViewById(R.id.main_tab_layout);
        tLayout.setupWithViewPager(viewPager);
        tLayout.setTabMode(TabLayout.MODE_FIXED); initiate(); return view;
    private void initiate() {
        kontrolData.setKontrol(pref); kontrolData.readProfil();
    }
    @Override public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState); setHasOptionsMenu(true);
    }
    @Override
    public void onCreateOptionsMenu(Menu menu, MenuInflater inflater) {
        inflater.inflate(R.menu.menu_pilih_bulan, menu);
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        if (item.getItemId() == R.id.menu_bulan) {
            MonthSelectorDialog monthSelectorDialog = new
            MonthSelectorDialog();
            monthSelectorDialog.prepare(fragmentManager,
            R.id.nav_pembelian, kontrolData);
            monthSelectorDialog.show(fragmentManager, "");
        }
        return true;
    }

```

Modul Program 4.2.15 PembelianActivity

```

public class PembelianAdapter extends FragmentStatePagerAdapter {
    private static int PAGE_COUNT=2;
    private String[] pembelianAdapter = {"GRAFIK PEMBELIAN", "DETAIL
    PEMBELIAN"};
    public PembelianAdapter(FragmentManager fm){super(fm);}
    @Override
    public Fragment getItem(int position){
        switch(position){
            case 0: return new GrafikPembelian();
            case 1: return new DetailPembelian();
        }
        return null;
    }
    @Override public int getCount(){ return PAGE_COUNT;}
    @Override
    public CharSequence getPageTitle(int position){
        return pembelianAdapter[position];
    }
}

```

Modul Program 4.2.16 PembelianAdapter

```

public class PembelianAPI {
    @SerializedName("status") @Expose private String status;
    @SerializedName("period") @Expose private String period;
    @SerializedName("ringkasan") @Expose private Ringkasan ringkasan;
    @SerializedName("bahan_masuk") @Expose
    private List<BahanMasuk> bahanMasuk = null;
    public String getStatus() {return status;}
    public void setPeriod(String period) { this.period = period;}
}

```

```

public String getPeriod() { return period;}
public void setStatus(String status) { this.status = status;}
public Ringkasan getRingkasan() {return ringkasan;}
public void setRingkasan(Ringkasan ringkasan) {
    this.ringkasan = ringkasan;}
public List<BahanMasuk> getBahanMasuk() {return bahanMasuk;}
public void setBahanMasuk(List<BahanMasuk> bahanMasuk) {
    this.bahanMasuk = bahanMasuk;}}

```

Modul Program 4.2.17 PembelianAPI

```

public class DaftarTransaksi {
    @SerializedName("nm_stok")@Expose private String nmStok;
    @SerializedName("jumlah")@Expose private String jumlah;
    @SerializedName("nm_satuan")@Expose private String nmSatuan;
    @SerializedName("tanggal_beli") @Expose private String tanggalBeli;
    @SerializedName("harga") @Expose private String harga;
    @SerializedName("name") @Expose private String name;
    public String getNmStok() { return nmStok;}
    public void setNmStok(String nmStok) { this.nmStok = nmStok; }
    public String getJumlah() { return jumlah;}
    public void setJumlah(String jumlah) { this.jumlah = jumlah; }
    public String getNmSatuan() { return nmSatuan;}
    public void setNmSatuan(String nmSatuan) { this.nmSatuan = nmSatuan; }
    public String getTanggalBeli() { return tanggalBeli;}
    public void setTanggalBeli(String tanggalBeli) {
        this.tanggalBeli = tanggalBeli;}
    public String getHarga() { return harga;}
    public void setHarga(String harga) { this.harga = harga;}
    public String getName() { return name; }
    public void setName(String name) { this.name = name;}}

```

Modul Program 4.2.18 DaftarTransaksi

```

public class BahanMasuk {
    @SerializedName("id_master_stok")@Expose private String idMasterStok;
    @SerializedName("transaksi")@Expose private String transaksi;
    @SerializedName("nm_stok")@Expose private String nmStok;
    @SerializedName("jumlah")@Expose private String jumlah;
    @SerializedName("nm_satuan")@Expose private String nmSatuan;
    @SerializedName("harga")@Expose private String harga;
    @SerializedName("daftar_transaksi") @Expose
    private List<DaftarTransaksi> daftarTransaksi = null;
    public String getIdMasterStok() { return idMasterStok;}
    public void setIdMasterStok(String idMasterStok) {this.idMasterStok =
idMasterStok;}public String getTransaksi() {return transaksi;}
    public void setTransaksi(String transaksi){this.transaksi = transaksi;}
    public String getNmStok() {return nmStok;}
    public void setNmStok(String nmStok) {this.nmStok = nmStok;}
    public String getJumlah() { return jumlah;}
    public void setJumlah(String jumlah) { this.jumlah = jumlah;}
    public String getHarga() {return harga;}
    public String getNmSatuan() {return nmSatuan;}
    public void setNmSatuan(String nmSatuan) {this.nmSatuan = nmSatuan;}
    public List<DaftarTransaksi> getDaftarTransaksi() {
return daftarTransaksi;}
    public void setDaftarTransaksi(List<DaftarTransaksi> daftarTransaksi) {
        this.daftarTransaksi = daftarTransaksi;}
    public void setHarga(String harga) {this.harga = harga;}}

```

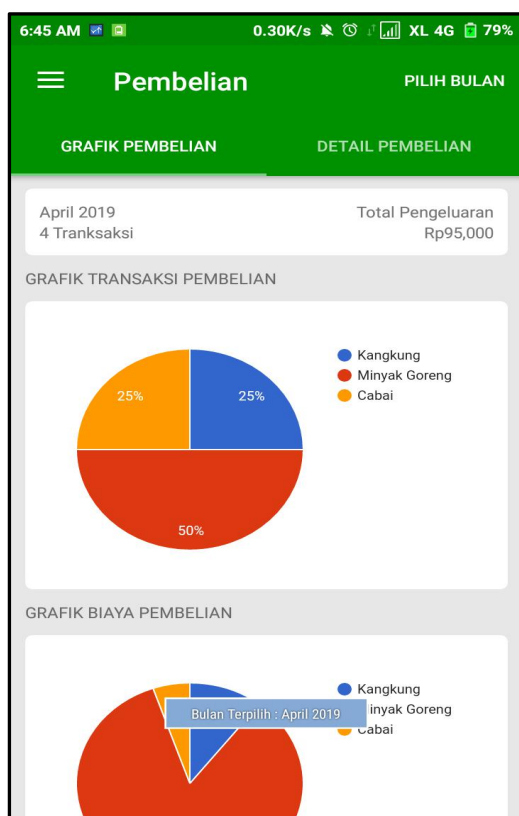
Modul Program 4.2.19 BahanMasuk

```

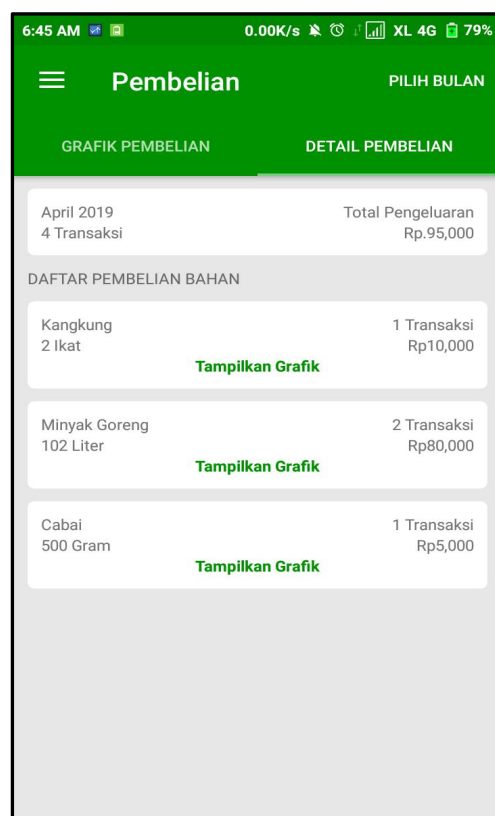
public class Ringkasan {
    @SerializedName("transaksi")@Expose private String transaksi;
    @SerializedName("pengeluaran")@Expose private String pengeluaran;
    @SerializedName("bulan") @Exposeprivate String bulan;
    public String getTransaksi() {return transaksi;}
    public void setTransaksi(String transaksi) {this.transaksi = transaksi;}
    public String getPengeluaran() { return pengeluaran; }
    public void setPengeluaran(String pengeluaran) {
        this.pengeluaran = pengeluaran;}
    public String getBulan() {return bulan;}
    public void setBulan(String bulan) {this.bulan = bulan;}}

```

Modul Program 4.2.20 Ringkasan



Gambar 4.2.7 Tampilan Grafik Pembelian



Gambar 4.2.8 Tampilan Detail Pembelian

7. Bagian Laporan Inventaris

Bagian ini menampilkan fitur untuk melihat laporan inventaris barang dapur modulnya adalah InventarisActivity, KategoriAPI, Kategori, DaftarInventaris, ScanResultActivity, dan ScanInventarisAPI.

```

public class InventarisActivity extends Fragment {
    private AppCompatActivity activity; private FragmentManager fragmentManager;
    public InventarisActivity() {} @Override
    public View onCreateView(@NonNull LayoutInflater inflater, ViewGroup
    container, Bundle savedInstanceState) {
        activity = (AppCompatActivity) getActivity();
        fragmentManager = getFragmentManager();
    }

```

```

        View view = inflater.inflate(R.layout.fragment_view_main, container,
false);

fragmentManager.beginTransaction().replace(R.id.fragment_view_main,    new
KategoriInventaris()).commit();
        Toolbar toolbar = view.findViewById(R.id.toolbar);
activity.supportActionBar(toolbar);activity.setTitle("Inventaris");
        DrawerLayout drawer = activity.findViewById(R.id.drawer_layout);
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
            activity, drawer, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close);
        drawer.addDrawerListener(toggle);toggle.syncState();return view;}
@Override
public void onCreate(@Nullable Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);setHasOptionsMenu(true);}
@Override
public void onCreateOptionsMenu(Menu menu, MenuInflater inflater) {
    inflater.inflate(R.menu.menu_inventaris_scan, menu);}
@Override public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if(id == R.id.action_scan){
        IntentIntegrator integrator =
IntentIntegrator.forSupportFragment(this);
integrator.setCaptureActivity(AnyOrientationCaptureActivity.class);
        integrator.setPrompt("Arahkan kamera pada BARCODE/QRCORE");
        integrator.setBeepEnabled(false);
        integrator.setOrientationLocked(false);integrator.initiateScan();}
        return super.onOptionsItemSelected(item);}
@Override public void onActivityResult(int requestCode, int resultCode,
Intent data) {
    IntentResult result =
IntentIntegrator.parseActivityResult(requestCode, resultCode, data);
    if(result != null) {
        if(result.getContents() == null) {
            Toast.makeText(activity, "Cancelled",
Toast.LENGTH_LONG).show();
        } else {pindah(result.getContents());}
        } else {super.onActivityResult(requestCode, resultCode, data);}}
private void pindah(String result){Bundle bundle = new Bundle();
    Toast.makeText(activity, "Hasil" + result,
Toast.LENGTH_LONG).show();
    bundle.putString("ID_BARANG", result);
    Fragment frag = new ScanResultActivity();frag.setArguments(bundle);
    fragmentManager.beginTransaction().replace(R.id.fragment_view_main,
frag).addToBackStack("Inventaris").commit();}}

```

Modul Program 4.2.21 InventarisActivity

```

public class KategoriAPI {
    @SerializedName("status")@Expose private String status;
    @SerializedName("kategori")@Expose private List<Kategori> data = null;
    public String getStatus() { return status;}
    public void setStatus(String status) {this.status = status;}
    public List<Kategori> getData() { return data;}
    public void setData(List<Kategori> data) { this.data = data;}
}

```

Modul Program 4.2.22 KategoriAPI


```

public class Kategori {
    @SerializedName("id_kategori")@Expose private String idKategori;
    @SerializedName("nm_kategori")@Expose private String nmKategori;
    @SerializedName("jumlah")@Expose private String jumlah;
    @SerializedName("total_harga")@Expose private Object totalHarga;
    @SerializedName("daftar_inventaris") @Expose
    private List<DaftarInventaris> daftarInventaris = null;
    public String getIdKategori() { return idKategori;}
    public void setIdKategori(String idKategori) {
        this.idKategori = idKategori;}
    public String getNmKategori() { return nmKategori;}
    public void setNmKategori(String nmKategori) {
        this.nmKategori = nmKategori;}
    public String getJumlah() { return jumlah;}
    public void setJumlah(String jumlah) { this.jumlah = jumlah;}
    public Object getTotalHarga() { return totalHarga;}
    public void setTotalHarga(Object totalHarga) {
        this.totalHarga = totalHarga;}
    public List<DaftarInventaris> getDaftarInventaris() {
        return daftarInventaris;}
    public void setDaftarInventaris(List<DaftarInventaris>
    daftarInventaris) {this.daftarInventaris = daftarInventaris;}
}

```

Modul Program 4.2.23 Kategori

```

public class DaftarInventaris {
    @SerializedName("id_barang")@Expose private String idBarang;
    @SerializedName("nm_barang")@Expose private String nmBarang;
    @SerializedName("no_seri")@Expose private String no_seri;
    @SerializedName("merk")@Expose private String merk;
    @SerializedName("warna")@Expose private String warna;
    @SerializedName("keadaan")@Expose private String keadaan;
    @SerializedName("harga")@Expose private String harga;
    public String getIdBarang() { return idBarang;}
    public void setIdBarang(String idBarang) { this.idBarang = idBarang;}
    public String getMerk() { return merk;}
    public void setMerk(String merk) { this.merk = merk;}
    public String getWarna() { return warna;}
    public void setWarna(String warna) { this.warna = warna;}
    public String getKeadaan() { return keadaan;}
    public void setKeadaan(String keadaan) { this.keadaan = keadaan;}
    public String getHarga() { return harga;}
    public void setHarga(String harga) {this.harga = harga;}
    public void setNmBarang(String nmBarang) { this.nmBarang = nmBarang;}
    public void setNo_seri(String no_seri) { this.no_seri = no_seri;}
    public String getNmBarang() { return nmBarang;}
    public String getNo_seri() { return no_seri;}}

```

Modul Program 4.2.24 DaftarInventaris

```

public class ScanResultActivity extends Fragment {
    private AppCompatActivity activity;private SharedPreferences
    pref;private KontrolData kontrolData;
    private TextView Nama, ID_Barang, Merk, Asal, Tgl, Warna, Harga, Kondisi,
    Lokasi;
    private String id_barang = "", APIKEY="";
    private FragmentManager fragmentManager;
    public ScanResultActivity() { }
    @Override

```

```

    public View onCreateView(@NonNull LayoutInflater inflater, ViewGroup
container,Bundle savedInstanceState) {        activity = (AppCompatActivity)
getActivity();
        fragmentManager = getFragmentManager();
        View                view                =
inflater.inflate(R.layout.fragment_inventaris_scan_result,        container,
false);
        Nama = view.findViewById(R.id.scan_result_nama_barang);
        ID_Barang = view.findViewById(R.id.scan_result_id_barang);
        Merk = view.findViewById(R.id.scan_result_merk_barang);
        Asal = view.findViewById(R.id.scan_result_asal_barang);
        Tgl = view.findViewById(R.id.scan_result_tanggal_barang);
        Warna = view.findViewById(R.id.scan_result_warna_barang);
        Harga = view.findViewById(R.id.scan_result_harga_barang);
        Kondisi = view.findViewById(R.id.scan_result_keadaan_barang);
        Lokasi = view.findViewById(R.id.scan_result_lokasi_barang);
        TextView Btn_Kondisi = view.findViewById(R.id.scan_result_btn_keadaan);
        activity.setTitle("Hasil Scan");
        pref = activity.getSharedPreferences("akun", MODE_PRIVATE);
        kontrolData = new KontrolData();initiate();
        if(getArguments() != null){Bundle bundle = getArguments();
            id_barang = bundle.getString("ID_BARANG", "");loaderAPI();}
        return view;}
    private void initiate(){kontrolData.setKontrol(pref);
        kontrolData.readProfil();APIKEY = kontrolData.getApi_key();}}
    private void loaderAPI(){
        final LoadingDialog loadingDialog = new LoadingDialog(getContext());
        final RuntimeDialog runtimeDialog = new RuntimeDialog();
        final FormatTanggal formatTanggal = new FormatTanggal();
        loadingDialog.setPesan(getString(R.string.dialog_load_data));
        loadingDialog.show();
        runtimeDialog.setTitle(getString(R.string.dialog_load_fail_title));
        runtimeDialog.setMsg(getString(R.string.dialog_load_fail_msg));
        Service service = Client.getClient().create(Service.class);
        Call<ScanInventarisAPI> call = service.getInventaris(id_barang, APIKEY);
        call.enqueue(new Callback<ScanInventarisAPI>() {@Override
            public void onResponse(Call<ScanInventarisAPI> call,
Response<ScanInventarisAPI> response) {
                if(response.body() != null){
                    String status = response.body().getStatus();loadingDialog.dismiss();
                    if(status.equals("success")){
                        Inventaris                inventaris                =
response.body().getInventaris();
                        if(!inventaris.getIdBarangMasuk().equals("null")){
                            Nama.setText(inventaris.getNmBarang());
                            ID_Barang.setText(inventaris.getIdBarangMasuk());
                            Merk.setText(inventaris.getMerk());
                            Asal.setText(inventaris.getAsal());
                            Tgl.setText(formatTanggal.format(inventaris.getTanggalBeli()));
                            Warna.setText(inventaris.getWarna());
                            String harga = "Rp."+inventaris.getHarga();
                            Harga.setText(harga);
                            Kondisi.setText(inventaris.getKeadaan());
                            String lokasi = inventaris.getNmRuang()+" "+inventaris.getNmGedung();
                            Lokasi.setText(lokasi);
                        }
                    }else{runtimeDialog.show(fragmentManager, "");}}}
        @Override
        public void onFailure(Call<ScanInventarisAPI> call, Throwable t)

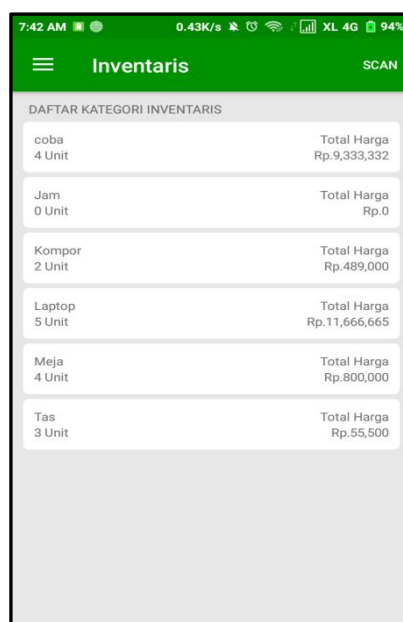
```

```
loadingDialog.dismiss();runtimeDialog.show(fragmentManager, "");}}});}
}
```

Modul Program 4.2.25 ScanResultActivity

```
public class ScanInventarisAPI {
    @SerializedName("status")@Exposeprivate String status;
    @SerializedName("inventaris")@Expose
    private Inventaris inventaris = null;
    public String getStatus() { return status;}
    public void setStatus(String status) { this.status = status;}
    public Inventaris getInventaris() { return inventaris; }
    public void setInventaris(Inventaris inventaris) {
        this.inventaris = inventaris;}
}
```

Modul Program 4.2.26 ScanInventarisAPI



Gambar 4.2.9 Tampilan Inventaris

8. Bagian Laporan Kunjungan

Bagian ini menampilkan fitur untuk melihat laporan pembelian kunjungan dapur modulnya adalah KunjunganActivity, KunjunganAdapter, KunjunganAPI, Kunjungan, DaftarTgl, DaftarPengunjung dan Ranking.

```
public class KunjunganActivity extends Fragment {
    private FragmentManager fragmentManager;
    private KontrolData kontrolData;private SharedPreferences pref;
    public KunjunganActivity() { }
    @Override public View onCreateView(@NonNull LayoutInflater inflater,
    ViewGroup container, Bundle savedInstanceState) {
        AppCompatActivity activity = (AppCompatActivity) getActivity();
        View view = inflater.inflate(R.layout.fragment_view_pager_main,
        container, false);
        ViewPager vPager = view.findViewById(R.id.main_viewpager);
```

```

        Toolbar toolbar = view.findViewById(R.id.toolbar);
        if(activity!=null){
            pref = activity.getSharedPreferences("akun", MODE_PRIVATE);
            kontrolData = new KontrolData();
            fragmentManager = getFragmentManager();
            activity.setSupportActionBar(toolbar);activity.setTitle("Kunjungan");
            DrawerLayout drawer = activity.findViewById(R.id.drawer_layout);
            ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
                activity, drawer, toolbar,
                R.string.navigation_drawer_open, R.string.navigation_drawer_close);
            drawer.addDrawerListener(toggle);toggle.syncState();}
            KunjunganAdapter      kunjunganAdapter      =      new
            KunjunganAdapter(getFragmentManager());
            vPager.setAdapter(kunjunganAdapter);
            TabLayout tLayout = view.findViewById(R.id.main_tab_layout);
            tLayout.setupWithViewPager(vPager); initiate();return view;}
            private void initiate(){kontrolData.setKontrol(pref);
            kontrolData.readProfil();}
            @Override public void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);setHasOptionsMenu(true);}
            @Override
            public void onCreateOptionsMenu(Menu menu, MenuInflater inflater) {
                inflater.inflate(R.menu.menu_pilih_bulan, menu);}
            @Override
            public boolean onOptionsItemSelected(MenuItem item) {
                if(item.getItemId() == R.id.menu_bulan){
                    MonthSelectorDialog monthSelectorDialog = new MonthSelectorDialog();
                    monthSelectorDialog.prepare(fragmentManager,
                    R.id.nav_kunjungan, kontrolData);
                    monthSelectorDialog.show(fragmentManager, "");}
                return true;}}

```

Modul Program 4.2.27 KunjunganActivity

```

public class KunjunganAdapter extends FragmentStatePagerAdapter {
    private static int PAGE_COUNT=3;
    private String[] kunjunganAdapter = {"GRAFIK KUNJUNGAN", "RAKING
    KUNJUNGAN", "DETAIL KUNJUNGAN"};
    private String kunjungan;private Bundle bundle = new Bundle();
    public void setBundle(String kunjungan) {this.kunjungan = kunjungan;}
    public KunjunganAdapter(FragmentManager fm){ super(fm);}
    @Override
    public Fragment getItem(int position){
        switch(position){
            case 0: return new GrafikKunjungan();
            case 1: return new RankKunjungan();
            case 2: return new DetailKunjungan();
        } return null; }
    @Override
    public int getCount(){ return PAGE_COUNT; }
    @Override
    public CharSequence getPageTitle(int position){
        return kunjunganAdapter[position];}}

```

Modul Program 4.2.28 KunjunganAdapter

```

public class KunjunganAPI {
    @SerializedName("status")@Expose private String status;
    @SerializedName("msg") @Expose private String msg;
    @SerializedName("kunjungan") @Expose private Kunjungan kunjungan;

```

```

@SerializedName("ranking")@Expose
private List<Ranking> ranking = null;
public String getStatus() { return status; }
public void setStatus(String status) { this.status = status; }
public String getMsg() { return msg; }
public void setMsg(String msg) { this.msg = msg; }
public Kunjungan getKunjungan() { return kunjungan; }
public void setKunjungan(Kunjungan kunjungan) {
    this.kunjungan = kunjungan; }
public List<Ranking> getRanking() { return ranking; }
public void setRanking(List<Ranking> ranking) { this.ranking = ranking; }
}

```

Modul Program 4.2.29 KunjunganAPI

```

public class Kunjungan implements Serializable {
    @SerializedName("jumlah")@Expose private Integer jumlah;
    @SerializedName("daftar_tgl") @Expose
    private List<DaftarTgl> daftarTgl = null;
    public Integer getJumlah() { return jumlah; }
    public void setJumlah(Integer jumlah) { this.jumlah = jumlah; }
    public List<DaftarTgl> getDaftarTgl() { return daftarTgl; }
    public void setDaftarTgl(List<DaftarTgl> daftarTgl) {
        this.daftarTgl = daftarTgl; }
}

```

Modul Program 4.2.30 Kunjungan

```

public class DaftarTgl {
    @SerializedName("jumlah")@Expose private String jumlah;
    @SerializedName("tgl")@Expose private String tgl;
    @SerializedName("daftar_pengunjung")@Expose
    private List<DaftarPengunjung> daftarPengunjung = null;
    public String getJumlah() { return jumlah; }
    public void setJumlah(String jumlah) { this.jumlah = jumlah; }
    public String getTgl() { return tgl; }
    public void setTgl(String tgl) { this.tgl = tgl; }
    public List<DaftarPengunjung> getDaftarPengunjung() {
        return daftarPengunjung; }
    public void setDaftarPengunjung(List<DaftarPengunjung>
    daftarPengunjung) {
        this.daftarPengunjung = daftarPengunjung; }
}

```

Modul Program 4.2.31 DaftarTgl

```

public class DaftarPengunjung {
    @SerializedName("id_karyawan")@Expose private String idKaryawan;
    @SerializedName("nm_karyawan")@Expose private String nmKaryawan;
    @SerializedName("tgl")@Expose private String tgl;
    public String getIdKaryawan() { return idKaryawan; }
    public void setIdKaryawan(String idKaryawan) {
        this.idKaryawan = idKaryawan; }
    public String getNmKaryawan() { return nmKaryawan; }
    public void setNmKaryawan(String nmKaryawan) {
        this.nmKaryawan = nmKaryawan; }
    public String getTgl() { return tgl; }
    public void setTgl(String tgl) { this.tgl = tgl; }
}

```

Modul Program 4.2.32 DaftarPengunjung

```

public class Ranking {
    @SerializedName("nm_karyawan")@Exposeprivate String nmKaryawan;
    @SerializedName("jumlah")@Exposeprivate String jumlah;
    public String getNmKaryawan() { return nmKaryawan;}
    public void setNmKaryawan(String nmKaryawan) {
        this.nmKaryawan = nmKaryawan;}
    public String getJumlah() { return jumlah;}
    public void setJumlah(String jumlah) { this.jumlah = jumlah;}}

```

Modul Program 4.2.33 Ranking



Gambar 4.2.10 Tampilan Grafik Kunjungan

RANK KUNJUNGAN DAPUR TELADAN

RANK	Nama Karyawan	Kunjungan
1	Udin Hazard	25 Kali
2	Andika Yuda	22 Kali
3	Allison Becker	21 Kali
4	Indra Safri	20 Kali
5	Jandla Eka	19 Kali
6	Dwi Irianto	17 Kali
7	Vigit Waluyo	16 Kali
8	Chaton Satria	14 Kali
9	Gusti Randa	10 Kali

Gambar 4.2.11 Tampilan Ranking Kunjungan

DETAIL DAFTAR PENGUNJUNG MEI 2019

Tanggal	Pengunjung
Kamis, 02 Mei 2019	6 Orang
Rabu, 01 Mei 2019	5 Orang

Gambar 4.2.12 Tampilan Daftar Kunjungan

9. Bagian Pengaturan Akun

Bagian ini menampilkan fitur untuk melihat informasi akun dan merubah kata sandi modulnya adalah AkunProfilku, ProfilAPI dan Profil.

```
public class AkunProfilku extends Fragment {
    private SharedPreferences pref;private KontrolData kontrolData;
    private Profil profil;private AppCompatActivity activity;
    private TextView Nama, Email, HakAkses;
    private EditText NewPW, ConfrimPW, OldPW; private Button Simpan;
    private LinearLayout viewProfil;
    public AkunProfilku() {    }
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, ViewGroup
container,Bundle savedInstanceState) {
        activity = (AppCompatActivity) getActivity();
        View view = inflater.inflate(R.layout.fragment_akun_profilku,
container, false);
        Toolbar toolbar = view.findViewById(R.id.toolbar);
        activity.supportActionBar(toolbar);
        activity.setTitle("Pengaturan Akun"); setHasOptionsMenu(true);
        DrawerLayout drawer = activity.findViewById(R.id.drawer_layout);
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
            activity, drawer, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close);
        drawer.addDrawerListener(toggle);toggle.syncState();
        Nama = view.findViewById(R.id.profil_nama);
        Email = view.findViewById(R.id.profil_email);
        HakAkses = view.findViewById(R.id.profil_hakakses);
        viewProfil = view.findViewById(R.id.profil_view);
        NewPW = view.findViewById(R.id.profil_newpassword);
        ConfrimPW = view.findViewById(R.id.profil_confirm_newpassword);
        OldPW = view.findViewById(R.id.profil_oldpassword);
        Simpan = view.findViewById(R.id.akun_profilku_simpan);
        viewProfil.setVisibility(View.GONE);
        pref = activity.getSharedPreferences("akun", MODE_PRIVATE);
        kontrolData = new KontrolData();
        kontrolData.setKontrol(pref);
        kontrolData.readProfil();profil = new Profil();loaderAPI();
        Simpan.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String newPW = NewPW.getText().toString(),
                    confrimPW = ConfrimPW.getText().toString(),
                    oldPW = OldPW.getText().toString();
                if(newPW.equals("") || confrimPW.equals("") ||
oldPW.equals("")){if(newPW.equals("")){
                    NewPW.setError("Belum Terisi!");NewPW.requestFocus();}
if(confrimPW.equals("")){ConfrimPW.setError("Belum
Terisi!");ConfrimPW.requestFocus();}if(oldPW.equals("")){
                    OldPW.setError("Belum Terisi!");    OldPW.requestFocus();}
                }else if(!newPW.equals(confrimPW)){
                    ConfrimPW.setError("Masukan tidak sama dengan Kata Sandi Baru!");
                    ConfrimPW.requestFocus();}else{
                    gantiPWAPI(kontrolData.getUser_id(), oldPW, newPW);}}});
```

```

        return view;}
    private void gantiPWAPI(String id, String password, String newpassword){
        final LoadingDialog loadingDialog = new LoadingDialog(getContext());
        final RuntimeDialog runtimeDialog = new RuntimeDialog();
        loadingDialog.setPesan("Merubah kata sandi...");
        loadingDialog.show();
        runtimeDialog.setTitle(getString(R.string.dialog_load_fail_title));
        runtimeDialog.setMsg(getString(R.string.dialog_load_fail_msg));
        Service service = Client.getClient().create(Service.class);
        Call<GantiPWAPI> call = service.gantiPW(id, password, newpassword,
        getAPIKEY());
        call.enqueue(new Callback<GantiPWAPI>() {
            @Override
            public void onResponse(Call<GantiPWAPI> call,
            Response<GantiPWAPI> response) {String status;
                if(response.body() != null){
                    status = response.body().getStatus(); loadingDialog.dismiss();
                    if(status.equals("succes")){
                        kontrolData.setPassword(NewPW.getText().toString());
                        kontrolData.writeProfil();
                        runtimeDialog.setTitle("Berhasil!");
                        runtimeDialog.setMsg("Kata Sandi berhasil dirubah");
                        runtimeDialog.show(activity.getSupportFragmentManager(), "");
                    }else{
                        runtimeDialog.setTitle("Gagal!");
                        runtimeDialog.setMsg("Kata Sandi gagal dirubah");
                        runtimeDialog.show(activity.getSupportFragmentManager(), "");}}}
            @Override
            public void onFailure(Call<GantiPWAPI> call, Throwable t) {
                loadingDialog.dismiss();
                runtimeDialog.show(getChildFragmentManager(), "");}}});
    private void loaderAPI(){
        final LoadingDialog loadingDialog = new LoadingDialog(getContext());
        final RuntimeDialog runtimeDialog = new RuntimeDialog();
        loadingDialog.setPesan(getString(R.string.dialog_load_data));
        loadingDialog.show();
        runtimeDialog.setTitle(getString(R.string.dialog_load_fail_title));
        runtimeDialog.setMsg(getString(R.string.dialog_load_fail_msg));
        Service service = Client.getClient().create(Service.class);
        Call<ProfilAPI> call = service.getProfilku(getAPIKEY());
        call.enqueue(new Callback<ProfilAPI>() { @Override
            public void onResponse(Call<ProfilAPI> call, Response<ProfilAPI>
            response) { String status;
                if(response.body() != null){
                    status = response.body().getStatus();
                    loadingDialog.dismiss();
                    if(status.equals("success")){
                        profil = response.body().getProfil();
                        Nama.setText(profil.getName());
                        Email.setText(profil.getEmail());
                        HakAkses.setText(profil.getRoleName().toUpperCase(Locale.getDefault()));
                    }}else{runtimeDialog.show(activity.getSupportFragmentManager(), "");}
                    viewProfil.setVisibility(View.VISIBLE);}
            @Override
            public void onFailure(Call<ProfilAPI> call, Throwable t) {
                Toast.makeText(getContext(), "Failed to load",
                Toast.LENGTH_LONG).show();
                loadingDialog.dismiss();
                runtimeDialog.show(getChildFragmentManager(), "");
                viewProfil.setVisibility(View.VISIBLE); }}});

```



```
private String getAPIKEY(){ return kontrolData.getApi_key(); }
@Override
public void onCreateOptionsMenu(Menu menu, MenuInflater inflater) {
    super.onCreateOptionsMenu(menu, inflater); menu.clear();
    inflater.inflate(R.menu.menu_manajemen_akun_profil, menu);}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if(id == R.id.action_quit_akun){
        kontrolData.setState(0);kontrolData.writeProfil();
        startActivity(new Intent(getApplicationContext(), LoginActivity.class));
    } return super.onOptionsItemSelected(item); }}
```

Modul Program 4.2.34 AkunProfilku

```
public class ProfilAPI {
    @SerializedName("status")@Expose private String status;
    @SerializedName("profil") @Expose private Profil profil;
    public String getStatus() { return status; }
    public void setStatus(String status) { this.status = status; }
    public Profil getProfil() { return profil; }
    public void setProfil(Profil profil) { this.profil = profil; }
}
```

Modul Program 4.2.35 ProfilAPI

```
public class Profil {
    @SerializedName("name")@Exposeprivate String name;
    @SerializedName("email") @Expose private String email;
    @SerializedName("role_name")@Expose private String roleName;
    public String getName() { return name; }
    public void setName(String name) { this.name = name; }
    public String getEmail() { return email; }
    public void setEmail(String email) { this.email = email; }
    public String getRoleName() { return roleName; }
    public void setRoleName(String roleName) { this.roleName = roleName; }
}
```

Modul Program 4.2.36 Profil

Gambar 4.2.13 Tampilan Pengaturan Akun

10. Bagian Manajemen Akun

Bagian ini menampilkan fitur untuk melihat informasi akun dan merubah kata sandi modulnya adalah ManajemenList, ManajemenTambah, TambahAPI dan UsersAPI.

```
public class ManajemenList extends Fragment {
    private RecyclerView recyclerView; private KontrolData kontrolData;
    private AppCompatActivity activity;
    private List<Users> usersList = new ArrayList<>();
    private FragmentManager fragmentManager;
    public ManajemenList() { } @Override
    public View onCreateView(@NonNull LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
        fragmentManager = getFragmentManager();
        activity = (AppCompatActivity) getActivity();
        View view = inflater.inflate(R.layout.fragment_akun_manajemen_list,
container, false);
        Toolbar toolbar = view.findViewById(R.id.toolbar_akun);
        activity.supportActionBar(toolbar);
        activity.setTitle("Daftar Pengguna");
        DrawerLayout drawer = activity.findViewById(R.id.drawer_layout);
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
            activity, drawer, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close);
        drawer.addDrawerListener(toggle);
        toggle.syncState(); setHasOptionsMenu(true);
        recyclerView = view.findViewById(R.id.users_daftar_akun);
        SharedPreferences pref = activity.getSharedPreferences("akun",
MODE_PRIVATE);
        kontrolData = new KontrolData(); kontrolData.setKontrol(pref);
        kontrolData.readProfil(); loaderAPI();
        RecyclerView.LayoutManager layoutManager = new
LinearLayoutManager(getContext());
        recyclerView.setLayoutManager(layoutManager);
        recyclerView.setItemAnimator(new DefaultItemAnimator()); return
view;}
    private void loaderAPI() {
        final LoadingDialog loadingDialog = new LoadingDialog(getContext());
        final RuntimeDialog runtimeDialog = new RuntimeDialog();
        loadingDialog.setPesan(getString(R.string.dialog_load_data));
        loadingDialog.show();
        runtimeDialog.setTitle(getString(R.string.dialog_load_fail_title));
        runtimeDialog.setMsg(getString(R.string.dialog_load_fail_msg));
        Service service = Client.getClient().create(Service.class);
        Call<UsersAPI> call = service.getUsers(getAPIKEY());
        call.enqueue(new Callback<UsersAPI>() {
            @Override public void onResponse(Call<UsersAPI> call,
Response<UsersAPI> response) { String status; if(response.body() != null){
                status = response.body().getStatus(); loadingDialog.dismiss();
                if(status.equals("success")){
                    usersList = response.body().getUsers();
                    UsersListAdapter usersListAdapter = new
UsersListAdapter(usersList, kontrolData.getUser_id(), getContext(),
getChildFragmentManager(), kontrolData.getApi_key());
                    recyclerView.setAdapter(usersListAdapter)} } else{
                runtimeDialog.show(activity.supportFragmentManager(), "");
            }
        });
    }
}
```

```

        } } @Override
        public void onFailure(Call<UsersAPI> call, Throwable t) {
            Toast.makeText(getContext(), "Failed to load"+t.getMessage(),
Toast.LENGTH_LONG).show();
            loadingDialog.dismiss();
            runtimeDialog.setMsg(t.getMessage());
            runtimeDialog.show(getChildFragmentManager(), "");
        } }
        private String getAPIKEY(){ return kontrolData.getApi_key(); }
        @Override public void onCreateOptionsMenu(Menu menu, MenuInflater
inflater) {super.onCreateOptionsMenu(menu, inflater);
            inflater.inflate(R.menu.menu_manajemen_akun_tambah, menu); }
        @Override public boolean onOptionsItemSelected(MenuItem item) {
            int id = item.getItemId();
            if(id == R.id.action_tambah_akun){
                fragmentManager.beginTransaction().replace(R.id.content_main,
new ManajemenTambah()).addToBackStack("manajemen").commit(); }
            return super.onOptionsItemSelected(item); } }

```

Modul Program 4.2.37 Manajemen List

```

public class ManajemenTambah extends Fragment {
    private KontrolData kontrolData; private Spinner daftar_hakakses;
    private AppCompatActivity activity; private EditText Nama, Email;
    public ManajemenTambah() { } @Override
    public View onCreateView(@NonNull LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
        activity = (AppCompatActivity) getActivity();
        View view = inflater.inflate(R.layout.fragment_akun_manajemen_tambah,
container, false); Button Tambah =
view.findViewById(R.id.akun_manajemen_tambah);
        daftar_hakakses = view.findViewById(R.id.akun_manajemen_spin_hakakses);
        Nama = view.findViewById(R.id.akun_manajemen_nama);
        Email = view.findViewById(R.id.akun_manajemen_email);
        Toolbar toolbar = view.findViewById(R.id.toolbar_tambah);
        activity.supportActionBar(toolbar);
        activity.setTitle("Daftar Pengguna");
        DrawerLayout drawer = activity.findViewById(R.id.drawer_layout);
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
            activity, drawer, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close);
        drawer.addDrawerListener(toggle); toggle.syncState();
        final ArrayAdapter<CharSequence> spinlength =
ArrayAdapter.createFromResource(activity.getContext(),
R.array.daftar_hakakses, android.R.layout.simple_spinner_dropdown_item);
        spinlength.setDropDownViewResource(android.R.layout.simple_spinner_dropdo
wn_item); daftar_hakakses.setAdapter(spinlength);
        SharedPreferences pref = activity.getSharedPreferences("akun",
MODE_PRIVATE);
        kontrolData = new KontrolData(); kontrolData.setKontrol(pref);
        kontrolData.readProfil();
        Tambah.setOnClickListener(new View.OnClickListener() {
            @Override public void onClick(View v) { prepareSave(); } });
        return view; }
    private void prepareSave() {
        String temp = daftar_hakakses.getSelectedItem().toString(),
            hak = "", nama = Nama.getText().toString(),
            email = Email.getText().toString();
        switch (temp) {

```

```

        case "ADMIN": hak = "2"; break;
        case "KEPALA DAPUR": hak = "3"; break;
        case "JURU MASAK": hak = "4"; break;
        case "VIP": hak = "1"; break;}
    if(nama.equals("") || email.equals("")){
        if(nama.equals("")){ Nama.setError("Nama Belum Diisi!");
        Nama.requestFocus();} if(email.equals("")){
            Email.setError("Email Belum Diisi!"); Email.requestFocus();}
        }else{ addAPI(nama, email, hak); } }
    private void addAPI(String nama, final String email,String hak){
        final LoadingDialog loadingDialog = new LoadingDialog(getContext());
        final RuntimeDialog runtimeDialog = new RuntimeDialog();
        loadingDialog.setPesan("Menambahkan..."); loadingDialog.show();
        runtimeDialog.setTitle(getString(R.string.dialog_load_fail_title));
        runtimeDialog.setMsg(getString(R.string.dialog_load_fail_msg));
        Service service = Client.getClient().create(Service.class);
        Call<TambahAPI> call = service.tambahPengguna(nama, email, hak,
        getAPIKEY());
        call.enqueue(new Callback<TambahAPI>() { @Override
            public void onResponse(Call<TambahAPI> call, Response<TambahAPI>
            response) { String status;
                if(response.body() != null){
                    status = response.body().getStatus(); loadingDialog.dismiss();
                    if(status.equals("succes")){
                        runtimeDialog.setTitle("Berhasil!");
                        runtimeDialog.setMsg(email+" berhasil ditambahkan");
                        runtimeDialog.show(activity.getSupportFragmentManager(), "");
                    }else{
                        runtimeDialog.setTitle("Gagal!");
                        runtimeDialog.setMsg(email+" gagal ditambahkan");
                        runtimeDialog.show(activity.getSupportFragmentManager(), ""); } }}
            @Override
            public void onFailure(Call<TambahAPI> call, Throwable t) {
                loadingDialog.dismiss();
                runtimeDialog.show(getChildFragmentManager(), ""); } })); }
    private String getAPIKEY(){ return kontrolData.getApi_key(); }}

```

Modul Program 4.2.38 Manajemen Tambah

```

public class TambahAPI {
    @SerializedName("status") @Expose private String status;
    public String getStatus() { return status; }
    public void setStatus(String status) { this.status = status; }
}

```

Modul Program 4.2.39 TambahAPI

```

public class UsersAPI {
    @SerializedName("status") @Expose private String status;
    @SerializedName("users") @Expose private List<Users> users = null;
    public String getStatus() { return status; }
    public void setStatus(String status) { this.status = status; }
    public List<Users> getUsers() { return users; }
    public void setUsers(List<Users> users) { this.users = users; }
}

```

Modul Program 4.2.40 UserAPI

Gambar 4.2.14 Tampilan Manajemen List

Gambar 4.2.15 Tampilan Tambah Pengguna

11. Bagian Kontrol Data

Bagian ini menampilkan fitur untuk kontrol terhadap data saat sesi login, modulnya adalah Akun dan KontrolData.

```
public class Akun{
    private FormatTanggal formatTanggal = new FormatTanggal();
    private String api_key = "" , nama = "", email = "", role_id = "",
        user_id = "", password = "";
    private int state = 0;
    public void setSemua(String email, String password, String api_key, int
state, String nama, String user_id, String role_id){
        this.email = email; this.api_key = api_key;
        this.state = state; this.password = password; this.nama = nama;
        this.user_id = user_id; this.role_id = role_id;
    }
    public void setApi_key(String api_key) { this.api_key = api_key; }
    public void setEmail(String email) { this.email = email; }
    public void setPassword(String password) { this.password = password; }
    public String getPassword() { return password; }
    public void setState(int state) { this.state = state }
    public int getState() { return state; }
    public String getEmail() { return email }
    public String getApi_key() { return api_key; }
    public void setNama(String nama) { this.nama = nama; }
    public String setBulan_sekarang() {
        Calendar calendar = Calendar.getInstance();
        int month = calendar.get(Calendar.MONTH) + 1;
        int year = calendar.get(Calendar.YEAR);
    }
}
```

```

        String bulan;if(month<10){ bulan = "0" + month + "-" + year; }else{
            bulan = month + "-" + year; } return bulan; }
    public void setRole_id(String role_id) { this.role_id = role_id; }
    public void setUser_id(String user_id) { this.user_id = user_id; }
    public String getNama() { return nama; }
    public String getBulan_sekarang() { return setBulan_sekarang(); }
    public String getFormat_BulanSekarang(){
        return formatTanggal.formatBulan(setBulan_sekarang()); }
    public String getRole_id() { return role_id; }
    public String getUser_id() { return user_id; }
}

```

Modul Program 4.2.41 Akun

```

public class KontrolData extends Akun {
    private SharedPreferences pref;private SharedPreferences.Editor prefEdit;
    private Gson gson = new Gson();
    private final String Email_KEY = "EMAIL", PW_KEY = "PW", API_KEY = "API",
        State_KEY = "STATE", NAMA_KEY = "NAMA",USERID_KEY = "ID",
        ROLEID_KEY = "ROLE", TGL_KEY="TGL",RKS_KEY="RKS",BMK_KEY="BMK",
        RANK_KEY="RANK",PMK_KEY="PMK",PMBL_KEY="PMBL";
    public void setKontrol(SharedPreferences pref) { this.pref = pref; }
    public void writeProfil(){ prefEdit = pref.edit();
        prefEdit.putString(Email_KEY, getEmail());
        prefEdit.putString(PW_KEY, getPassword());
        prefEdit.putString(API_KEY, getApi_key());
        prefEdit.putInt(State_KEY, getState());
        prefEdit.putString(NAMA_KEY, getNama());
        prefEdit.putString(USERID_KEY, getUser_id());
        prefEdit.putString(ROLEID_KEY, getRole_id()); prefEdit.apply();}
    public void readProfil(){
        String Email = pref.getString(Email_KEY, ""),
            password = pref.getString(PW_KEY, ""),
            API = pref.getString(API_KEY, ""),
            Nama = pref.getString(NAMA_KEY, ""),
            UserID = pref.getString(USERID_KEY, ""),
            RoleID = pref.getString(ROLEID_KEY, "");
        int State = pref.getInt(State_KEY, 0);
        setSemua(Email, password, API, State, Nama, UserID, RoleID);
        setBulan_sekarang(); }
    public void writeKunjungan(List<DaftarTgl> daftarTgl, List<Ranking>
daftarRanking){
        String dftrTgl = gson.toJson(daftarTgl);
        String dftrRank = gson.toJson(daftarRanking);
        Log.d("TAG","dftrTgl = " + dftrTgl);
        Log.d("TAG","dftrRank = " + dftrRank); prefEdit = pref.edit();
        prefEdit.putString(TGL_KEY, dftrTgl);
        prefEdit.putString(RANK_KEY, dftrRank); prefEdit.apply(); }
    public List<DaftarTgl> getDaftarTgl(){
        List<DaftarTgl> daftarTgl;
        String response = pref.getString(TGL_KEY, "");
        daftarTgl = gson.fromJson(response, new }
    public void writePembelian(PembelianAPI pembelianAPI){
        String pembelian = gson.toJson(pembelianAPI);
        prefEdit = pref.edit();
        prefEdit.putString(PMBL_KEY, pembelian); prefEdit.apply();}
    public PembelianAPI getPembelian(){ PembelianAPI pembelianAPI;
        String response = pref.getString(PMBL_KEY, "");
        pembelianAPI = gson.fromJson(response, new

```

```

TypeToken<PembelianAPI>(){}.getType());
    return pembelianAPI; }
    public void writePembelian(Ringkasan ringkasanIn, List<BahanMasuk>
bahanMasuk){
        String rks = gson.toJson(ringkasanIn),
        bmk = gson.toJson(bahanMasuk); prefEdit = pref.edit();
        prefEdit.putString(RKS_KEY, rks); prefEdit.putString(BMK_KEY, bmk);
        prefEdit.apply();}
    public Ringkasan getRingkasan(){Ringkasan ringkasan;
        String response = pref.getString(RKS_KEY, "");
        ringkasan = gson.fromJson(response, new
TypeToken<Ringkasan>(){}.getType());
        return ringkasan; }
    public List<BahanMasuk> getBahanMasuk(){
        List<BahanMasuk> bahanMasuk;
        String response = pref.getString(BMK_KEY, "");
        bahanMasuk = gson.fromJson(response, new
TypeToken<List<BahanMasuk>>(){}.getType());
        return bahanMasuk;}
    public void writePemakaian(PemakaianAPI pemakaian){
        String pmk = gson.toJson(pemakaian); prefEdit = pref.edit();
        prefEdit.putString(PMK_KEY, pmk); prefEdit.apply();}
    public PemakaianAPI getPemakaian(){ PemakaianAPI pemakaian;
        String response = pref.getString(PMK_KEY, ""); pemakaian =
gson.fromJson(response, new TypeToken<PemakaianAPI>(){}.getType());
        return pemakaian;}}

```

Modul Program 4.2.42 KontrolData

BAB V

PENUTUP

Berdasarkan hasil dari pembahasan dan data-data yang diperoleh dari kerja praktek di Yayasan Sinai Indonesia, maka penulis dapat mengambil kesimpulan dan saran yang mungkin berguna bagi pihak instansi dalam menghadapi permasalahan yang dihadapi.

5.1 Kesimpulan

Kesimpulan yang bisa diambil setelah menjalankan kerja praktek di Yayasan Sinai Indonesia yaitu *Cloud Computing* dapat diintegrasikan dengan perangkat smartphone android dengan protokol RestfulAPI yang menggunakan format pertukaran data JSON .

5.2 Saran

1. *Data caching* pada android dapat di optimalkan lagi untuk mengurangi beban web service
2. Penambahan galeri foto untuk inventaris dapur

DAFTAR PUSTAKA

- Antonio Gamez-Diaz, Pablo Fernandez, Antonio Ruiz-Cortes.(2017). Universidad de Sevilla. *An analysis of RESTful APIs offerings in the industry*.
- Anubhav Jain , Manoj Kumar, Anil Lambha.(2014). International Journal of Computer Applications. *An Overview and Trends in Cloud Computing*.
- Harjit Singh.(2012). International Journal of Electronics and Computer Science Engineering . *Current Trends in Cloud Computing A Survey of Cloud Computing Systems*.
- SmartBear. n.d. *REST 101:The Beginner's Guide to Using and Testing RESTful APIs*.
- Wu-Lung Huang, Kuang-Min Wu, Miao-sheng Chen, Chih-Ming Chen.(2011). e-CASE & e-Tech International Conference. *The Study of Using QR Code in the Mobile Tourist Guide Map*.