LAMPIRAN

Listing Program

```
private fun login() {
    val email =
binding.tieEmail.text.toString()
    val password =
binding.tiePassword.text.toString()
    when {
        email.isEmpty() -> {
            binding.tilEmail.error = "Email
tidak boleh kosong"
            binding.tieEmail.requestFocus()
        }
        !email.matches("[a-zA-Z0-9._-]+@[a-
zA-Z0-9.-]+\\.[a-zA-Z]{2,}".toRegex()) -> {
            binding.tilEmail.error = "Format
email salah"
            binding.tieEmail.requestFocus()
        }
        else -> {
            binding.tilEmail.error = null
        }
    }
    if (password.isEmpty()) {
        binding.tilLoginPassword.error =
"Password tidak boleh kosong"
        binding.tiePassword.requestFocus()
        return
    } else {
        binding.tilLoginPassword.error =
null
    }
```

```
var emailValid = false
private fun validEmail(): String? {
    val emailText =
binding.etEmail.text.toString()
    if (emailText.isEmpty()) {
        binding.tilEmail.error = "Email
tidak boleh kosong"
        emailValid = false
    } else if (!emailText.matches("[a-zA-Z0-
9._-]+@[a-zA-Z0-9.-]+\\.[a-zA-
Z]{2,}".toRegex())) {
        binding.tilEmail.error = "Format
Email Tidak Sesuai"
        emailValid = false

    } else {
        binding.tilEmail.error = null
        emailValid = true

    }
    return null
}
```

```
val adapter = DataExpertAdapter {
dataExpertModel ->
        val action =
DataExpertFragmentDirections.actionDataExper
tFragmentToDetailDataExpertFragment2(
                dataExpertModel.typePaket
            )
        action.selectedTypePaket =
dataExpertModel.typePaket
        findNavController().navigate(action)
    }

    binding.rvDataExpert.adapter = adapter
    binding.rvDataExpert.layoutManager =
LinearLayoutManager(requireContext())

    val dataExpertList = listOf(
        DataExpertModel(
            "1",
            "Dasar",
            "Maksimal 5 permintaan query",
            "Pengerjaan 1 hari",
            "Visualisasi data",
            "Laporan eye catching",
            "Pengerjaan tepat waktu",
            "Rp. 49.900,00"
        ),
        DataExpertModel(
            "2",
            "Standar",
            "Maksimal 10 permintaan query",
            "Pengerjaan 1 hari",
            "Visualisasi data",
            "Laporan eye catching",
            "Pengerjaan tepat waktu",
            "Rp. 89.900,00"
        ),
        DataExpertModel(
            "3",
            "Premium",
            "Maksimal 20 permintaan query",
            "Pengerjaan 1 hari",
            "Visualisasi data",
            "Laporan eye catching",
            "Pengerjaan tepat waktu",
            "Rp. 159.900,00"
        )
    )
    adapter.submitList(dataExpertList)
}
```

```
val selectedTypePaket =

DetailDataExpertFragmentArgs.fromBundle(requ
ireArguments()).selectedTypePaket
Log.d("DetailDataExpertFrag", "Selected
typePaket: $selectedTypePaket")

activity?.title = "Detail Paket Data Expert"
with(binding) {
    ivDown.setOnClickListener {
```

```
        llListReport.visibility =
View.VISIBLE
        ivDown.isVisible = false
        ivUp.isVisible = true
    }
    ivUp.setOnClickListener {
        llListReport.visibility = View.GONE
        ivDown.isVisible = true
        ivUp.isVisible = false
    }
    var maxEditTexts = 0
    when (selectedTypePaket) {
        "Dasar" -> {
            paket = "dasar"
            activity?.title = "Detail Paket
Dasar"
            tvDetail.text =
                "Pembelian sekali bayar
untuk bisa mendapatkan 5 permintaan query.
Lebih banyak permintaan akan semakin hemat
dan untung banyak!"
            tvDescription1.text = "Bebas 5
Permintaan Query"
            tvDescription2.text =
"Pengerjaan dalam 1 hari"
            tvTotalHarga.text = "Rp.
49.900,00"
            tvDetailHarga2.text = "Rp.
49.900,00"
            tvBiayaAdmin.text = "0"
            totalAmount = 49900.0
            maxEditTexts = 5
        }
        "Standar" -> {
            paket = "standar"
            activity?.title = "Detail Paket
Standar"
            tvDetail.text =
                "Pembelian sekali bayar
untuk bisa mendapatkan 10 permintaan query.
Lebih banyak permintaan akan semakin hemat
dan untung banyak!"
            tvDescription1.text = "Bebas 10
Permintaan Query"
            tvDescription2.text =
"Pengerjaan dalam 1 hari"
            tvTotalHarga.text = "Rp.
89.900,00"
            tvDetailHarga2.text = "Rp.
89.900,00"
            tvBiayaAdmin.text = "0"
            totalAmount = 89900.0
            maxEditTexts = 10
        }
        "Premium" -> {
            paket = "premium"
            activity?.title = "Detail Paket
Premium"
            tvDetail.text =
                "Pembelian sekali bayar
untuk bisa mendapatkan 20 permintaan query.
Lebih banyak permintaan akan semakin hemat
dan untung banyak!"
            tvDescription1.text = "Bebas 20
Permintaan Query"
            tvDescription2.text =
"Pengerjaan dalam 1 hari"
            tvTotalHarga.text = "Rp.
159.900,00"
            tvDetailHarga2.text = "Rp.
159.900,00"
            tvBiayaAdmin.text = "0"
            totalAmount = 189900.0
            maxEditTexts = 20
        }
    }

    addBtn.setOnClickListener {
        if (editTextCount <= maxEditTexts) {
            val newTextInputLayout =
TextInputLayout(requireContext())

newTextInputLayout.boxBackgroundColor =

ContextCompat.getColor(context!!,
android.R.color.white)

newTextInputLayout.boxBackgroundMode =

TextInputLayout.BOX_BACKGROUND_OUTLINE

newTextInputLayout.setBoxCornerRadii(

resources.getDimension(R.dimen.dimen_4),

resources.getDimension(R.dimen.dimen_4),

resources.getDimension(R.dimen.dimen_4),

resources.getDimension(R.dimen.dimen_4)
            )

newTextInputLayout.isHintAnimationEnabled =
true

            val editText =
TextInputEditText(newTextInputLayout.context
)
            newTextInputLayout.id =
View.generateViewId()
            newTextInputLayout.layoutParams
= LinearLayout.LayoutParams(

LinearLayout.LayoutParams.MATCH_PARENT,

LinearLayout.LayoutParams.WRAP_CONTENT
            ).apply {
                topMargin =

resources.getDimensionPixelSize(R.dimen.list
_item_spacing_half)
                bottomMargin =

resources.getDimensionPixelSize(R.dimen.list
_item_spacing_half)
            }
            editText.hint = "Permintaan
$editTextCount"


newTextInputLayout.addView(editText)

linearlayout.addView(newTextInputLayout)
```

```
            editTextCount++
        }
    }


fun getNotification(id:Int) =
api.getNotifications(id)



@GET("notifications/user/{id}")
fun getNotifications(
    @Path("id") id: Int,
    @Header("Authorization") authorization:
String = SAVED_TOKEN
): Call<NotificationResponse>



import
com.google.gson.annotations.SerializedName

data class NotificationResponse(
    @SerializedName("data")
    val `data`: Data,
    @SerializedName("message")
    val message: String,
    @SerializedName("status")
    val status: Int
)

class Differ :
DiffUtil.ItemCallback<Notification>() {
    override fun areItemsTheSame(oldItem:
Notification, newItem: Notification):
Boolean {
        return oldItem == newItem
    }

    override fun areContentsTheSame(oldItem:
Notification, newItem: Notification):
Boolean {
        return oldItem.hashCode() ==
newItem.hashCode()
    }
}
```