|  |
| --- |
| API Nominative (Personally Identifiable Information) |
| --Personally Identifiable Information (PII)  with PII as (  select    denk.id as id,    AEAD.DECRYPT\_STRING(denk.keyset,den.nominative\_identity\_number, den.id) as ktp\_petani,    --AEAD.DECRYPT\_STRING digunakan untuk mendecrypt PII atau informasi identitas    -- 1=denk.keyset mengacu ke keyset dari table decrypt. misal kita mendecrypt dim\_ecofin\_nominatif, berarti harus join ke table p2p\_access\_enigma.dim\_ecofin\_nominatif\_keys    -- 2=mengacu ke column yang ingin didecrypt atau dibuka datanya, Misal kita ingin mendecrypt nominative\_name, maka parameter ke 2 ini diisi dengan den.nominative\_name    -- 3=mengacu ke support key untuk parameter 2, contohnya kita ingin men-decrypt dim\_ecofin\_nominatif.nominative\_name di sheet itu sudah distate bahwa support keynya adalah id.  Sehingga parameter ke-3 diisi dengan den.id    AEAD.DECRYPT\_STRING(denk.keyset,den.nominative\_name, den.id) as farmer\_name,    AEAD.DECRYPT\_STRING(denk.keyset,den.spouse\_identity\_number, den.id) as ktp\_spouse,    AEAD.DECRYPT\_STRING(denk.keyset,den.spouse\_name, den.id) as spouse\_name  from    `alami-group-data.p2p\_access.dim\_ecofin\_nominatif` as den  left join    `alami-group-data.p2p\_access\_enigma.dim\_ecofin\_nominatif\_keys` as denk  on den.id = denk.id --on pada join menyesuaikan dengan sheet pada P2P Access Layer Design  )  , line\_facility as  (  select    date(line\_facility\_sent\_time, "Asia/Jakarta") as approved\_time,    de.id as id,    df.id as id2,    status,    status\_message,    pefindo\_grade,    submission\_method,    lokasi\_efishery\_point,    line\_facility\_amount  from    `alami-group-data.p2p\_access.dim\_ecofin\_nominatif` as de  left join    `alami-group-data.p2p\_access.dim\_individual\_line\_facility` as df  on    de.individual\_id = df.individual\_id  where    (de.submission\_method="API" or de.submission\_method="Manual")  and    (status\_message="Line Facility Sent" or status\_message="Line Facility Signed")  and    (de.partner\_id ="2c91808274029f1c017402f293560005" or de.partner\_id ="2c918082835230120183547eb56906dd")    --jika tidak menggunakan tanda kurung, maka query setelah And yaitu status message akan dieksekusi terlebih dahulu    --jika menggunakan tanda kurung maka query dalam tanda kurung akan dieksekusi setelah "where = submission method=API"    --OR digunakan jika salah satu kondisi terpenuhi maka return the value    --AND digunakan dengan minimal 2 kondisi terpenuhi maka return the value, jika salah satu atau keduanya tidak terpenuhi maka akan Null/No  order by 1 desc  )  select    pii.id,    PII.ktp\_petani,    PII.farmer\_name,    PII.ktp\_spouse,    PII.spouse\_name,    lf.approved\_time,    lf.status,    lf.pefindo\_grade,    lf.submission\_method,    lf.lokasi\_efishery\_point,    max(lf.line\_facility\_amount) as latest\_lf,    min(lf.line\_facility\_amount) as lf\_before\_upgrading  from    PII  left join    line\_facility as lf  on PII.id = lf.id  where    lf.status is not null  group by 1,2,3,4,5,6,7,8,9,10  order by 3 |
| Basic SQL |
| select    loa\_no,    loan\_product\_type,    UPPER(coalesce(rm\_name,"unidentified")) as rm\_name,--coalesce digunakan untuk mengisi Null Value menjadi data yang diinginkan    lower(product\_type\_description) as product\_type\_description,    upper(campaign\_type) as campaign\_type,--capitalize huruf    disbursement\_amount,  from    `alami-group-data.p2p\_access.dim\_application`  where    loan\_product\_type not like ("Invoice Financing")    --not like digunakan untuk filter data yang ada pada satu kolom, tampilkan data kecuali untuk Invoice Financing  and    campaign\_type like "%lic%"    --like sama halnya "where campaign\_type = "public" namun bisa digunakan untuk memfilter yang lebih spesifk, seperti kata yang mengandung "lic"  and    disbursement\_amount >= 1000000000 --comparison operator greater than equal  and    rm\_name in ("Muhammad Rasyid Ridho") --IN digunakan untuk memfilter lebih spesifik pada data dalam satu kolom  order by 6 desc |
| Intermediate 1 |
|  |
| select    coalesce(tenor\_in\_days,"Unidentified") as tenor,    count(1) as transaksi,    count(distinct tenor\_in\_days) as unique\_tenor,    --cek ke unique-an data dalam satu kolom    sum(disbursement\_amount) as total\_disbursement,    max(disbursement\_amount) as max\_disbursement,    min(disbursement\_amount) as min\_disbursement,    round(avg(disbursement\_amount),-2) as avg\_disbursement\_1,    --round, -2 = jika bilangan negatif maka maka membulatkan desimal menjadi bilangan bulat, contoh = 667,1678910 menjadi 668,00    round(avg(disbursement\_amount),2) as avg\_disbursement\_2,    --round,2 = jika bilangan positif maka membulatkan nilai desimal 2 angka di belakang koma contoh : 667,1678910 menjadi 667,16    case      when count(1) >5000 and sum(disbursement\_amount)>50000000000 then "huge transaction"      when count(1) >1000 and sum(disbursement\_amount)>30000000000 then "medium\_transaction"      --case when digunakan untuk membuat kategori atas suatu kondisi, setelah when diikuti oleh kolom yang memiliki data discreate(Bilangan Bulat) atau continuous (Data yang didapat dari proses perhitungan)      --then adalah kategori yang diinginkan      else "average\_trx"      --else digunakan sebagai kondisi terakhir jika kedua kondisi 1 dan 2 sudah terpenuhi, mirip dengan value\_if\_false pada excel      end as kategori\_transaksi      --end as merupakan nama kolom  from    `alami-group-data.p2p\_access.dim\_ecofin\_application`  group by 1  having count(1) > 100 --having digunakan untuk memfilter aggregate function, jika tidak ada aggregate function maka bisa menggunakan where  order by 2 desc |
| Intermediate 2 |
| --outstanding LF principal+ujroh  select    lf.day,    de.name\_short,    sum(lf.outstanding\_amount) as principal,    round(sum(lf.outstanding\_mp\_fee\_amount),-2) as mp\_fee,    sum(lf.outstanding\_ujrah\_amount) as margin,    round(sum(lf.outstanding\_mp\_fee\_amount\*0.11),-2) as outstanding\_tax,    round(sum(lf.outstanding\_amount)+sum(lf.outstanding\_mp\_fee\_amount)+sum(lf.outstanding\_ujrah\_amount)+sum(lf.outstanding\_mp\_fee\_amount\*0.11),-2) as total\_tagihan  from    `alami-group-data.p2p\_access.dim\_company\_ecofin` as de  left join    `alami-group-data.p2p\_mart.summary\_application\_outstanding\_daily` as lf  on    de.company\_id = lf.company\_id  where    lf.day = current\_date("Asia/Jakarta")  group by 1,2  order by 7 desc |
| Advance 1 |
| select    date(application\_disbursed\_time) as disburse\_date,    date(application\_disbursed\_time, "Asia/Jakarta") as disburse\_date\_timezione,    time(application\_disbursed\_time) as disburse\_time,--without 24 hour    time(application\_disbursed\_time,"Asia/Jakarta") as disburse\_time\_timezone, --using indonesia time zone 24 Hour    cast(tenor\_in\_days AS INT64) as tenor, --cast is function to reform data type, tenor\_in\_day was string and reform to integer data/whole number/bilangan bulat    date\_add(date(application\_disbursed\_time), interval cast(tenor\_in\_days AS INT64) day) as maturity\_date, --date\_add digunakan untuk menambahkan tanggal, setelah interval harus bilangan bulat    date\_trunc(date(application\_disbursed\_time), month) as month1,--diambil tanggal pertama dari tiap bulan    extract(month from date(application\_disbursed\_time)) as  month2, --mengambil bulan pada tiap tanggal format "mm"    cast(date(application\_disbursed\_time) as string format "month") as month3,    date\_diff(date\_add(date(application\_disbursed\_time), interval cast(tenor\_in\_days AS INT64) day), date(application\_disbursed\_time), Day) as tenor\_in\_day2, --date\_diff to calculate aggregate from latest date to early date, dari tgl jatuh tempo ke tgl disburse selisih berapa hari?    coalesce(concat(total\_quantity," " ,order\_unit),"Unidentified") as quantity\_per\_unit,    length(id) as length\_id,    split(replace(app\_initial\_code, ".","-"),"-") as all\_loan\_id\_split, --untuk menggunakan multiple split perlu direplace terlebih dahulu dengan delimitter yang sama lalu displit    split(replace(app\_initial\_code, ".","-"),"-")[offset(0)] as EF\_split,--[offset,(0)] digunakan untuk membuat kolom baru disebelahnya, 0 jika kolom berada pada urutan pertama setelah kolom yang displit    split(replace(app\_initial\_code, ".","-"),"-")[offset(1)] as short\_name\_split, --[offset,(1)] digunakan untuk membuat kolom baru disebelahnya, 1 jika kolom berada pada urutan kedua setelah kolom yang displit, begitu seterusnya    (select      sum(disbursement\_amount)    from      `alami-group-data.p2p\_access.dim\_application`    where      loan\_product\_type="EF-Indirect"    and      (company\_id = "2c91808274029f1c017402f293560005" or company\_id = "2c918082835230120183547eb56906dd")) as total\_disburse\_MTN\_KBX --subqueries tidak bisa 1 single value, perlu aggregate function  from    `alami-group-data.p2p\_access.dim\_ecofin\_application`  where    application\_disbursed\_time is not null |
| Advance 2 |
| with PII as  (  select    da.client\_id as client\_id\_PII,    AEAD.DECRYPT\_STRING(dak.keyset,da.client\_name,da.app\_id) as farmer\_name  from    `alami-group-data.p2p\_access.dim\_application` as da  left join    `alami-group-data.p2p\_access\_enigma.dim\_application\_keys` as dak  on da.app\_id = dak.id  where    da.company\_id = "2c91808274029f1c017402f293560005" or da.company\_id = "2c918082835230120183547eb56906dd"  order by 2  )  , first\_trx as  (  select    client\_id as client\_id,    company\_name\_short,    min(date(disburse\_time, "Asia/Jakarta")) as first\_disburse  from    `alami-group-data.p2p\_access.dim\_application`  group by 1,2  )  select    date\_trunc(date(da.disburse\_time, "Asia/Jakarta"), month) as month,    da.company\_name\_short,    case when date\_trunc(date(da.disburse\_time, "Asia/Jakarta"), month) = date\_trunc(ft.first\_disburse,month) then "New Farmer" else "Repeat Farmer" end as Farmer\_Category,    count(distinct ft.client\_id) as total\_trx,    sum(da.disbursement\_amount) as total\_disburse  from    `alami-group-data.p2p\_access.dim\_application` as da  left join    first\_trx as ft  on da.client\_id = ft.client\_id  where    da.loan\_product\_type ="EF-Indirect"  and    (da.company\_id = "2c91808274029f1c017402f293560005" or da.company\_id = "2c918082835230120183547eb56906dd")  and    date\_trunc(date(disburse\_time, "Asia/Jakarta"), month) is not null  group by 1,2,3  order by 1 |
| Application Efishery 2023 |
| select    date(da.disburse\_time,"Asia/Jakarta") as disburse\_date,    da.app\_initial\_code as loan\_id,    dea.transaction\_number as order\_id,    da.loa\_no as loa\_number,    da.company\_name\_short as company,    da.tenor\_in\_days as tenor,    da.disbursement\_amount as disbursement,    da.portfolio\_status\_name,    da.maturity\_date,    date(da.repayment\_time, "Asia/Jakarta") as repayment\_date,    date\_diff(date(da.repayment\_time,"Asia/Jakarta"), da.maturity\_date, day) as dpd,    case      when date\_diff(date(da.repayment\_time,"Asia/Jakarta"), da.maturity\_date, day) = 0 then "On-time"      when date\_diff(date(da.repayment\_time,"Asia/Jakarta"), da.maturity\_date, day) < 0 then "early repayment"      when date\_diff(date(da.repayment\_time,"Asia/Jakarta"), da.maturity\_date, day) < 14 then "late 1-14 days"      when date\_diff(date(da.repayment\_time,"Asia/Jakarta"), da.maturity\_date, day) is null then "outstanding"    else "late >14 days"    end as dpd\_category  from    `alami-group-data.p2p\_access.dim\_application` as da  left join    `alami-group-data.p2p\_access.dim\_ecofin\_application` as dea  on    da.loa\_no = dea.loa\_number  where    loan\_product\_type = "EF-Indirect"  and    date(disburse\_time, "Asia/Jakarta") between "2023-01-01" and "2023-12-31"  and    (da.company\_id = "2c91808274029f1c017402f293560005" or da.company\_id = "2c918082835230120183547eb56906dd")  order by 1 |
| Commodities |
| select    commodities,    commodities\_trx,    disburse\_per\_commodity,    round((disburse\_per\_commodity/sum(disburse\_per\_commodity) over ())\*100,2) as ratio\_to\_disburse  from  (select    distinct en.commodity as commodities,    count(1) as commodities\_trx,    sum(ea.disbursement\_amount) as disburse\_per\_commodity  from    `alami-group-data.p2p\_access.dim\_ecofin\_nominatif` as en  right join    `alami-group-data.p2p\_access.dim\_ecofin\_application` as ea  on    en.individual\_id = ea.individual\_id  where    en.commodity is not null  group by 1  order by 2 desc)  order by 2 desc |
| Cross Join |
| --cross join berfungsi mengkombinasikan seluruh table  with ecosystem as  (  select    sum(disbursement\_amount) as total\_disbursement,    max(disbursement\_amount) as max\_disbursement,  from    `alami-group-data.p2p\_access.dim\_application`  where    loan\_product\_type="EF-Indirect"  )  select    da.product\_type\_description,    eco.\*  from    `alami-group-data.p2p\_access.dim\_application` as da  cross join ecosystem as eco  where    loan\_product\_type ="Invoice Financing" or loan\_product\_type = "PO Financing" |
| DPD Efishery |
| select    date\_trunc(date(repayment\_time,"Asia/Jakarta"),month) as repayment\_date,    company\_name\_short,    dpd\_category,    dpd\_category\_new,    count(loa\_no) as trx,    round(avg(dpd),2) as avg\_dpd,  from    `alami-group-data.p2p\_access.dim\_application`  WHERE loan\_product\_type='EF-Indirect'    AND repayment\_time IS NOT NULL    AND disburse\_time IS NOT NULL    AND portfolio\_status\_id='14'    AND (company\_name\_short="MTN" or company\_name\_short="KBX")  group by 1,2,3,4  order by 1 |
| Decrypt |
| SELECT      pn.id,      pn.partner\_full\_name,      pn.individual\_id,      pn.cif\_number,      AEAD.DECRYPT\_STRING(denk.keyset, pn.nominative\_name, pn.id) AS benef\_name,      AEAD.DECRYPT\_STRING(denk.keyset, pn.nominative\_phone\_number, pn.id) AS benef\_phone,      pn.email AS benef\_email,      AEAD.DECRYPT\_STRING(denk.keyset, pn.nominative\_identity\_number, pn.id) AS benef\_identity\_number,      pn.nominative\_npwp\_number,      pn.date\_of\_birth,      pn.place\_of\_birth,      pn.marital\_status,      pn.gender,      pn.job\_name,      pn.monthly\_income,      AEAD.DECRYPT\_STRING(denk.keyset, pn.spouse\_name, pn.id) AS spouse\_name,      AEAD.DECRYPT\_STRING(denk.keyset, pn.spouse\_identity\_number, pn.id) AS spouse\_identity\_number,  FROM `alami-group-data.p2p\_access.dim\_ecofin\_nominatif`  pn      LEFT JOIN `alami-group-data.p2p\_access\_enigma.dim\_ecofin\_nominatif\_keys` denk  ON denk.id = pn.id |
| Disburse Ecosystem |
| select    date\_trunc(date(disburse\_time,"Asia/Jakarta"), month) as disburse\_month,    sum(disbursement\_amount) as disbursement\_amount  from    `alami-group-data.p2p\_access.dim\_application`  where    loan\_product\_type="EF-Indirect"  and    date(disburse\_time, "Asia/Jakarta") between "2022-01-01" and "2022-12-31"  and    not company\_id = "2c9180828410bb5e018413363653066f" --not column\_name = "kategori" digunakan untuk pengecualian  group by 1  order by 1 desc |
| NoA Ecosystem |
| select    company\_name\_short,    date\_trunc(date(disburse\_time,"Asia/Jakarta"),month) as disburse\_date,    count(distinct client\_id) as NoA\_Unique,    sum(disbursement\_amount) as disbursement,  from    `alami-group-data.p2p\_access.dim\_application`  where    loan\_product\_type='EF-Indirect'  and    portfolio\_status\_id = "14"  and    date(disburse\_time,"Asia/Jakarta") between "2022-01-01" and "2023-01-31"  group by 1,2  order by 2 |
| Disburse MTN dan KBX |
| select    distinct dea.individual\_id,    count(1) as total\_transaksi,    sum(dea.disbursement\_amount) as total\_disbursement  from    `alami-group-data.p2p\_access.dim\_ecofin\_application` as dea  left join `alami-group-data.p2p\_access.dim\_individual\_line\_facility` as dlf on dea.individual\_id = dlf.individual\_id  where    dea.partner\_id = "2c91808274029f1c017402f293560005"  or    dea.partner\_id = "2c918082835230120183547eb56906dd"  group by 1  order by 2 desc |
| Example CTE (Common Table Expression) |
| with    funder\_first\_transaction as  (    select      funder\_id,      min(date(time\_of\_transaction, "Asia/Jakarta")) as first\_transaction\_date --mencari tanggap pertama chip-in setiap funder\_id    from      `alami-group-data.p2p\_access.fact\_transaction\_chipin`    where      transaction\_status="Accepted"    group by 1  )  --mencari data new dan repeat funder per bulan dengan salah satu sumber data di table yang sudah dibuat sebelumnya  select    date\_trunc(date(ftc.time\_of\_transaction, "Asia/Jakarta"), month) as transaction\_month, --tanggal pertama dalam tiap bulan    case when date\_trunc(date(time\_of\_transaction, "Asia/Jakarta"),month) =    date\_trunc(fft.first\_transaction\_date, month) then "New Funder" else "Repeat Funder" end as funder\_category,    --dengan kondisi ketika tanggal transaksi di table ftc sm dengan tgl transaksi di table funder frst transaction    count(distinct ftc.funder\_id) as total\_transaction\_funder  from    `alami-group-data.p2p\_access.fact\_transaction\_chipin` as ftc  left join    funder\_first\_transaction as fft on fft.funder\_id = ftc.funder\_id --lookup funder\_id di fft dan funder\_id di ftc  where    transaction\_status="Accepted"  group by 1,2  order by 1 desc, 2 |
| Homework 1 Advance 1 |
| select    payor\_name,    total\_application\_dpd\_0,    total\_application\_dpd\_1\_30,    total\_application\_dpd\_31\_60,    total\_application\_dpd\_61\_90  from  (select    upper(REGEXP\_REPLACE(TRIM(dc.client), r'\.|,|TBK', '')) as payor\_name,    count(case when dpd\_category\_new = "<=0" then app\_id else null end) as total\_application\_dpd\_0,    count(case when dpd\_category\_new = "1-30" then app\_id else null end) as total\_application\_dpd\_1\_30,    count(case when dpd\_category\_new = "31-60" then app\_id else null end) as total\_application\_dpd\_31\_60,    count(case when dpd\_category\_new = "61-90" then app\_id else null end) as total\_application\_dpd\_61\_90  from    `alami-group-data.p2p\_access.dim\_collateral` as dc  join    `alami-group-data.p2p\_access.dim\_application` as da  on    dc.application\_id = da.app\_id  where    dc.collateral\_type\_name="Invoice / Bill"  and    date(da.repayment\_time, "Asia/Jakarta") between "2021-01-01" and "2021-12-31"  group by 1)  order by  5 desc, 4 desc, 3 desc |
| Homework 2 Advance 1 |
| select    disburse\_month,    avg(disbursement\_sla\_day) avg\_disbursement\_sla\_day  from  (select    date\_trunc(date(ready\_to\_disbursement\_time, "Asia/Jakarta"), month) as disburse\_month,    date\_diff(date(disburse\_time,"Asia/Jakarta"), date(submission\_time,"Asia/Jakarta"), day) as disbursement\_sla\_day  from    `alami-group-data.p2p\_access.dim\_application`  where    date(disburse\_time) between "2022-01-01" and "2022-12-31"  )  group by 1  order by 1 |
| Window Function |
| --window function  --bisa digunakan sebagai alternatif group by  with window\_product as  (  select    distinct product\_type\_description as product\_type,  sum(disbursement\_amount) over (partition by product\_type\_description) as total\_disburse\_per\_product,    sum(disbursement\_amount\*alami\_margin\_pct/100) over (partition by product\_type\_description) as total\_mpf\_per\_product,  from    `alami-group-data.p2p\_access.dim\_application`  where    portfolio\_status\_id="14"  order by 2 desc  )  ,contribution as  (  select    wp.product\_type as product\_type,    wp.total\_disburse\_per\_product as disburse\_per\_product,    round(wp.total\_disburse\_per\_product/sum(da.disbursement\_amount),2)\*100 as contribution  from    `alami-group-data.p2p\_access.dim\_application` as da  cross join window\_product as wp  where    portfolio\_status\_id="14"  group by 1,2  )  , application as  (  select    date\_trunc(date(disburse\_time,"Asia/Jakarta"),month) as disburse\_date,    product\_type\_description as product,    sum(disbursement\_amount) as disbursement,    sum(disbursement\_amount\*alami\_margin\_pct/100) as mpf  from    `alami-group-data.p2p\_access.dim\_application`  where    portfolio\_status\_id="14"  and    disburse\_time is not null  group by 1,2  order by 1  )  select    ap.disburse\_date,    ap.product,    ap.disbursement,    sum(ap.disbursement) over (partition by ap.product order by ap.disburse\_date) as running\_total,    sum(ap.disbursement) over (partition by ap.disburse\_date) as month\_total,    sum(ap.mpf) over (partition by ap.disburse\_date) as mpf\_month\_total,    round((sum(ap.disbursement) / sum(ap.disbursement) over (partition by ap.disburse\_date))\*100,2) as contribution,    ap.mpf,    round((sum(ap.mpf) / sum(ap.mpf) over (partition by ap.disburse\_date))\*100,2) as mpf\_contribution,    --menghitung total disbursemen pada masing2 product pada bulan berjalan dibagi total disbursement pada bulan berjalan    --round((wp.total\_disburse\_per\_product/ sum (wp.total\_disburse\_per\_product) over (partition by ap.disburse\_date))\*100,2) contribution  from    application as ap  left join    window\_product as wp on ap.product = wp.product\_type  inner join    contribution as cb on wp.product\_type = cb.product\_type  group by 1,2,3,8  order by 1 |
| New Farmers/Customers |
| with ft as  (  select    client\_id as farmer,    date\_trunc(min(date(disburse\_time,"Asia/Jakarta")),month) as first\_trx  from    `alami-group-data.p2p\_access.dim\_application`  where    loan\_product\_type='EF-Indirect'  and    portfolio\_status\_id = "14"  group by 1  )  , new\_farmer as  (  select    date\_trunc(date(disburse\_time,"Asia/Jakarta"),Month) as month,    company\_name\_short,    count(distinct da.client\_id) as total\_farmer,    case when date\_trunc(date(da.disburse\_time,"Asia/Jakarta"),Month) = ft.first\_trx then "new farmer" else "repeat\_farmer" end as farmer\_category  from    `alami-group-data.p2p\_access.dim\_application` as da  left join    ft on ft.farmer = da.client\_id  where    da.loan\_product\_type='EF-Indirect'  and    da.portfolio\_status\_id = "14"  and    date(da.disburse\_time,"Asia/Jakarta") between "2022-01-01" and "2022-12-31"  and    (da.company\_name\_short = "MTN" or da.company\_name\_short = "KBX" )  group by 1,2,4  order by 1  )  select    \*  from    new\_farmer  where    farmer\_category = "new farmer" |
| Without Window Function |
| --menghitung contribusi partnership terhadap total disbursement di ALAMI  --tanpa window function  with partnership as  (  select    product\_type\_description,    sum(disbursement\_amount) as total\_disbursement\_partnership,  from    `alami-group-data.p2p\_access.dim\_application`  where    portfolio\_status\_id="14"  group by 1  order by 2 desc  )  ,total\_disbursement as  (  select    sum(disbursement\_amount) as total\_disbursement\_alami  from    `alami-group-data.p2p\_access.dim\_application`  where  portfolio\_status\_id="14"  )  select    partnership.product\_type\_description,    partnership.total\_disbursement\_partnership,    total\_disbursement.total\_disbursement\_alami,    round((partnership.total\_disbursement\_partnership/total\_disbursement.total\_disbursement\_alami)\*100,2) as contribution\_pct  from    partnership  cross join    total\_disbursement |
| PII Farmers |
| --Personally Identifiable Information (PII)  with PII as (  select    denk.id as id,    AEAD.DECRYPT\_STRING(denk.keyset,den.nominative\_identity\_number, den.id) as ktp\_petani,    --AEAD.DECRYPT\_STRING digunakan untuk mendecrypt PII atau informasi identitas    -- 1=denk.keyset mengacu ke keyset dari table decrypt. misal kita mendecrypt dim\_ecofin\_nominatif, berarti harus join ke table p2p\_access\_enigma.dim\_ecofin\_nominatif\_keys    -- 2=mengacu ke column yang ingin didecrypt atau dibuka datanya, Misal kita ingin mendecrypt nominative\_name, maka parameter ke 2 ini diisi dengan den.nominative\_name    -- 3=mengacu ke support key untuk parameter 2, contohnya kita ingin men-decrypt dim\_ecofin\_nominatif.nominative\_name di sheet itu sudah distate bahwa support keynya adalah id.  Sehingga parameter ke-3 diisi dengan den.id    AEAD.DECRYPT\_STRING(denk.keyset,den.nominative\_name, den.id) as farmer\_name,    AEAD.DECRYPT\_STRING(denk.keyset,den.spouse\_identity\_number, den.id) as ktp\_spouse,    AEAD.DECRYPT\_STRING(denk.keyset,den.spouse\_name, den.id) as spouse\_name  from    `alami-group-data.p2p\_access.dim\_ecofin\_nominatif` as den  left join    `alami-group-data.p2p\_access\_enigma.dim\_ecofin\_nominatif\_keys` as denk  on den.id = denk.id --on pada join menyesuaikan dengan sheet pada P2P Access Layer Design  )  , line\_facility as  (  select    date(line\_facility\_sent\_time, "Asia/Jakarta") as approved\_time,    date(created\_time, "Asia/Jakarta") as submission\_time,    de.id as id,    df.id as id2,    de.partner\_full\_name as company,    de.place\_of\_birth as tempat\_lahir,    de.date\_of\_birth as tanggal\_lahir,    de.kebutuhan\_pakan as kebutuhan\_pakan,    de.luas\_lahan\_kolam as luas\_kolam,    de.total\_hasil\_panen as hasil\_panen,    de.commodity as komoditas,    status,    status\_message,    pefindo\_grade,    submission\_method,    lokasi\_efishery\_point,    line\_facility\_amount  from    `alami-group-data.p2p\_access.dim\_ecofin\_nominatif` as de  left join    `alami-group-data.p2p\_access.dim\_individual\_line\_facility` as df  on    de.individual\_id = df.individual\_id  where    de.submission\_method="API" or de.submission\_method="Manual"  order by 1 desc  )  select    pii.id,    lf.company,    PII.ktp\_petani,    PII.farmer\_name,    PII.ktp\_spouse,    PII.spouse\_name,    lf.tempat\_lahir,    lf.tanggal\_lahir,    lf.submission\_time,    lf.approved\_time,    lf.status,    lf.pefindo\_grade,    lf.submission\_method,    lf.lokasi\_efishery\_point,    lf.luas\_kolam,    lf.komoditas,    lf.kebutuhan\_pakan,    lf.hasil\_panen,    max(lf.line\_facility\_amount) as latest\_lf,  from    PII  left join    line\_facility as lf  on PII.id = lf.id  where    lf.status is not null  group by 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18  order by 3 |
| Quiz Advance |
| --Total disbursement per bulan dari Repeat Benef  --Buat flagging untuk new benef dan repeat benef  --filter portofolio\_status\_id= "14"  with benef\_first\_disbursement as (  select    app\_id,    min(date(disbursement\_time,"Asia/Jakarta")) as first\_disbursement\_date  from    `alami-group-data.p2p\_access.dim\_application`  where    portfolio\_status\_id="14"  group by  1  )  select    date\_trunc(date(da.disbursement\_time,"Asia/Jakarta"),month) as disbursement\_month,    case when date\_trunc(date(disbursement\_time,"Asia/Jakarta"),month) = date\_trunc(date(bfd.first\_disbursement\_date),month) then "New Benef" else "Repeat Benef" end as benef\_category,    count(da.app\_id) as total\_disbursement\_benef  from    `alami-group-data.p2p\_access.dim\_application` as da  left join    benef\_first\_disbursement as bfd  on    bfd.app\_id = da.app\_id  where    case when date\_trunc(date(disbursement\_time,"Asia/Jakarta"),month) = date\_trunc(date(bfd.first\_disbursement\_date),month) then "New Benef" else "Repeat Benef" end  = "Repeat Benef"  group by    1,2 |
| Repayment |
| select    company\_name\_short,    date\_trunc(date(repayment\_time,"Asia/Jakarta"),month) as repayment\_date,    sum(disbursement\_amount) as repayment\_amount\_principal,    round(sum(disbursement\_amount\*ujrah\_pct/100),2) as margin,    round(sum(disbursement\_amount\*alami\_margin\_pct/100),2) as mp\_fee,    round((sum(disbursement\_amount\*alami\_margin\_pct/100) \* 11)/100,2) as tax,        round(sum(disbursement\_amount) +  round(sum(disbursement\_amount\*ujrah\_pct/100),2) + round(sum(disbursement\_amount\*alami\_margin\_pct/100),2) + round((sum(disbursement\_amount\*alami\_margin\_pct/100) \* 11)/100,2),2) as total\_tagihan  from    `alami-group-data.p2p\_access.dim\_application`  where    loan\_product\_type = "EF-Indirect"  and    date(repayment\_time,"Asia/Jakarta") between "2022-01-01" and "2023-01-31"  group by 1,2  order by 2 |
| Transaksi and Outstanding |
| with transaksi as (  select    distinct dea.individual\_id as farmer\_id,    dlf.line\_facility\_amount as lf\_amount,    count(distinct dea.individual\_id) as total\_transaksi,    sum(dea.disbursement\_amount) as total\_disbursement,    round((sum(dea.disbursement\_amount)/dlf.line\_facility\_amount)\*100,2) as utilisasi\_transaksi,  from    `alami-group-data.p2p\_access.dim\_ecofin\_application` as dea  left join    `alami-group-data.p2p\_access.dim\_individual\_line\_facility` as dlf on dea.individual\_id = dlf.individual\_id  where    (dea.partner\_id = "2c91808274029f1c017402f293560005"  or    dea.partner\_id = "2c918082835230120183547eb56906dd")  group by 1,2  order by 2 desc )  , outstanding as  (select    day as curren\_date,    company\_id,    client\_id,    loan\_product\_type,    replace(replace(rm\_team\_name, "Other","Ecosystem"),"Ecosystem Financing","Ecosystem") as rm\_team, --replace multiple string within one column    maturity\_date,    outstanding\_amount,    round(outstanding\_mp\_fee\_amount,0) as mp\_fee,    outstanding\_ujrah\_amount as margin,    round(outstanding\_mp\_fee\_amount\*0.11,0) as tax,    round(outstanding\_amount+outstanding\_mp\_fee\_amount+outstanding\_ujrah\_amount+outstanding\_mp\_fee\_amount\*0.11,0) as outstanding\_total  from    `alami-group-data.p2p\_mart.summary\_application\_outstanding\_daily`  where    day = current\_date("Asia/Jakarta")  and    loan\_product\_type = "EF-Indirect"  and    (company\_id = "2c91808274029f1c017402f293560005" or company\_id = "2c918082835230120183547eb56906dd")  )  select    trx.farmer\_id,    trx.lf\_amount,    trx.total\_transaksi,    trx.total\_disbursement,    trx.utilisasi\_transaksi,    os.outstanding\_amount  from    transaksi as trx  left join outstanding as os on trx.farmer\_id = os.client\_id |
| Unnest |
| select    application\_id,    id.invoice\_due  from    `alami-group-data.p2p\_access.dim\_collateral`,  UNNEST (invoice\_detail) as id |
| Utilization |
| -- utilization LF by transaction  -- ambil table disbursement dari dim\_application lalu filter ke loan\_product\_type = "EF-Indirect"  -- filter portofolio\_status\_id = 14  -- ambil kolom client\_id  -- transaksi petani perbulan  with farmer\_transaction as  (  select    client\_id,    date(disburse\_time,"Asia/Jakarta") as disburse\_date,    disbursement\_amount  from    `alami-group-data.p2p\_access.dim\_application`  where    loan\_product\_type = "EF-Indirect"  and    portfolio\_status\_id = "14"  order by 2 desc  )  -- farmer line facility  , farmer\_line\_facility as  (  select    day,    individual\_id,    line\_facility\_amount  from    `alami-group-data.p2p\_mart.summary\_individual\_line\_facility\_daily`  )  -- utilisasi dari seluruh transaksi/total LF  , utilization\_all\_trx as (  select    date\_trunc(ft.disburse\_date, month) as month\_disburse,    client\_id,    sum(ft.disbursement\_amount)/sum(flf.line\_facility\_amount) as all\_utilization\_lf\_trx,  from    farmer\_transaction as ft  left join    farmer\_line\_facility as flf  on    ft.client\_id = flf.individual\_id  group by 1,2  order by 1 desc )  -- final battle  select    month\_disburse,    sum(case when all\_utilization\_lf\_trx >= 0 and all\_utilization\_lf\_trx <= 0.25 then all\_utilization\_lf\_trx else 0 end ) as uti\_below\_25,    sum(case when all\_utilization\_lf\_trx > 0.25 and all\_utilization\_lf\_trx <= 0.50 then all\_utilization\_lf\_trx else 0 end) as uti\_25\_50,    sum(case when all\_utilization\_lf\_trx > 0.50 and all\_utilization\_lf\_trx <= 0.75 then all\_utilization\_lf\_trx else 0 end) as uti\_50\_75,    sum(case when all\_utilization\_lf\_trx > 0.75 and all\_utilization\_lf\_trx <= 1.00 then all\_utilization\_lf\_trx else 0 end) as uti\_75\_100,    sum(case when all\_utilization\_lf\_trx > 1.00 then all\_utilization\_lf\_trx else 0 end) as uti\_above\_100  from    utilization\_all\_trx  where    utilization\_all\_trx.month\_disburse is not null  group by 1  order by 1 |
| Windows Function 2 |
| --window function  --bisa digunakan sebagai alternatif group by  with window\_product as  (  select    distinct product\_type\_description as product\_type,    sum(disbursement\_amount) over (partition by product\_type\_description) as total\_disburse\_per\_product,    sum(disbursement\_amount\*alami\_margin\_pct/100) over (partition by product\_type\_description) as total\_mpf\_per\_product,  from    `alami-group-data.p2p\_access.dim\_application`  where    portfolio\_status\_id="14"  order by 2 desc  )  ,contribution as  (  select    wp.product\_type as product\_type,    wp.total\_disburse\_per\_product as disburse\_per\_product,    round(wp.total\_disburse\_per\_product/sum(da.disbursement\_amount),2)\*100 as contribution  from    `alami-group-data.p2p\_access.dim\_application` as da  cross join window\_product as wp  where    portfolio\_status\_id="14"  group by 1,2  )  , application as  (  select    date\_trunc(date(disburse\_time,"Asia/Jakarta"),month) as disburse\_date,    product\_type\_description as product,    sum(disbursement\_amount) as disbursement,    sum(disbursement\_amount\*alami\_margin\_pct/100) as mpf  from    `alami-group-data.p2p\_access.dim\_application`  where    portfolio\_status\_id="14"  and    disburse\_time is not null  group by 1,2  order by 1  )  select    ap.disburse\_date,    ap.product,    ap.disbursement,    sum(ap.disbursement) over (partition by ap.product order by ap.disburse\_date) as running\_total,    sum(ap.disbursement) over (partition by ap.disburse\_date) as month\_total,    round((sum(ap.disbursement) / sum(ap.disbursement) over (partition by ap.disburse\_date))\*100,2) as contribution,    ap.mpf,    round((sum(ap.mpf) / sum(ap.mpf) over (partition by ap.disburse\_date))\*100,2) as mpf\_contribution,    --menghitung total disbursemen pada masing2 product pada bulan berjalan dibagi total disbursement pada bulan berjalan    --round((wp.total\_disburse\_per\_product/ sum (wp.total\_disburse\_per\_product) over (partition by ap.disburse\_date))\*100,2) contribution  from    application as ap  left join    window\_product as wp on ap.product = wp.product\_type  inner join    contribution as cb on wp.product\_type = cb.product\_type  group by 1,2,3,7  order by 1 |
| Split |
| select    "iphone\_samsung-somi" as brand,    split(replace("iphone\_samsung-somi","-","\_"),"\_") as all\_split,    split(replace("iphone\_samsung-somi","-","\_"),"\_")[offset(0)] as split1,    split(replace("iphone\_samsung-somi","-","\_"),"\_")[offset(1)] as split2,    split(replace("iphone\_samsung-somi","-","\_"),"\_")[offset(2)] as split3, |